GEOSUPPORT™ SYSTEM

User Programming Guide

SOFTWARE VERSION 10.1

Department of City Planning / City of New York
Information Technology Division
Geographic Systems Section

SYSTEM

User Programming Guide

Software Version 10.1
January, 2007

TABLE OF CONTENTS

APPENDICES AND GLOSSARY

LIST OF TABLES

SUMMARY OF CHANGES AND NEW FEATURES

Version 9.7.5

- Support of Roadbed Request Switch

Version 10.0 / Version10.1

- Functionality is the same on all Geosupport platforms – viz. CICS, MVS Batch, and Desktop Edition. (All platforms produce compatible results).

- JCL changes are required in MVS batch environment. See Chapter VIII for details.

- The legacy version of Functions 1A/BL will not run as of Version 10.1. See Technical Bulletin 05-1 for further detail

- There are changes to Geosupport Messages and Geosupport Return Codes (GRCs). See Appendix 4 for the latest version of all messages and GRCs.

- Slight changes in Geosupport Work Area internal fields and filler fields were implemented. Since these fields are typically not referenced by a user application, these changes should be transparent to the user.

- Work Area1 Output fields – various fields are initialized to blanks to help avoid extraneous data in output fields.

- In a small percentage of cases, for Functions 1 and 1E, the spatial coordinates (X-Y coordinates) are somewhat different, but always lay along the same block face.

- When an input street name is not recognized, MVS Batch and Desktop Edition return ‘similar names’ as CICS did in previous versions.

- Browse functions (BB and BF) are now supported in MVS Batch and Desktop Edition in addition to CICS.

- New Features in Character–Only Work Areas (COW)
  - Function 3S Lion Node Number
  - B7SC (instead of B5SC) returned for browse functions, ‘similar names’, and cross streets.

- The GOAT utility on the mainframe has been enhanced to ‘remember’ user’s input.

CHAPTER I: SYSTEM OVERVIEW

I.1 Introduction

The Geosupport System is an integrated system of software and data files that processes New York City geographic locations. Input locations can be accepted in various forms, including addresses, place names, street intersections, block faces, street stretches and property parcels (tax lots). Geosupport standardizes and validates the input location and provides related geographic information, such as the community district, census tract and census block, zip code, tax block and tax lot, police precinct, cross streets, City Council district and spatial coordinates. In addition, Geosupport provides user-written applications with the means to retrieve data from the user's own files by geographic location in a consistent manner.

The Geosupport System was developed and is maintained as a service to all agencies of the City of New York by the staff of the Geographic Systems Section (GSS) of the Department of City Planning's Information Technology Division. GSS has been enhancing the Geosupport software and updating the system's data files continually since the system's introduction in 1983.

This document, the *Geosupport System User Programming Guide*, is a comprehensive technical description of the system and how to access it. The document is intended to be read primarily by technical users such as computer application designers, software analysts and programmers. Knowledge of IBM mainframe application programming is assumed.

Geosupport is installed on IBM mainframes at the city computer centers listed in Appendix 7, where it is used by over 30 city agencies as an integral component of many of their major data processing applications. An employee of each computer center's custodial agency, generally a systems programmer, has been designated as the Geosupport System Administrator (GSA) for that computer center. The GSA is responsible for installing or coordinating the installation of new Geosupport file releases and software versions received from GSS. At some computer centers, the GSA makes certain customizing modifications to Geosupport during installation, such as changing the data set names (DSNs) of Geosupport files to conform to local file naming conventions. The GSA is also the first responder for users encountering system-related Geosupport problems, and interacts with GSS staff as needed to resolve such problems. However, the GSA is not necessarily familiar with Geosupport from a user application perspective, and is not responsible for assisting users with application-related problems or design issues.

In general, Geosupport is accessible without restriction to anyone having a valid account at any of the computer centers where it is installed; no special passwords or security procedures are required beyond the normal data center logon procedures. In addition to running on IBM (and IBM-compatible) mainframes, GSS has developed Geosupport Desktop Edition for running in the Windows environment. There are also several methods by which Geosupport running on a city mainframe can be accessed by applications running on other hardware platforms, including 3270 emulation and access through the city's Intranet (e.g. GOAT on the Web).

For many user applications, only a subset of Geosupport's functions, features and data items are relevant. In addition, many Geosupport options have defaults which are appropriate for most applications. Thus, even though Geosupport is a large-scale, multi-feature system, the user effort required to design an application to access it is often relatively modest.

Geosupport has an Application Programming Interface (API) that enables it to be accessed directly from a user-written batch or CICS application program. (Geosupport cannot be run in the VM environment.)   The

Geosupport API supports application programs written in any programming language that can issue a standard IBM external program call. COBOL, PL/1, the various types of IBM mainframe assembler languages (hereafter referred to generically as 'Assembler'), NATURAL and C are five such languages. In addition, Geosupport has one batch and two interactive 'stand-alone' utility programs that enable users to satisfy many geographic processing needs without having to write custom application programs.

The ability of user-written applications to access Geosupport via its API enables users to avoid the burden of duplicating complex, specialized geographic processing routines within their own applications. The Geosupport API affords users total design control of their own applications, with their own input and output files, printed reports and screen formats; users develop those applications in their conventional data processing environments, using the programming languages they deem best suited to those applications. This contrasts with many geographic processing packages, such as Geographic Information System (GIS) products, which require users to develop their applications within the specialized environment of that package, often using a specialized, proprietary programming language and/or data base management system.

The Geosupport batch utility program is called the Geosupport Batch Address Translator (GBAT). GBAT requires no user programming; to run GBAT, the user simply sets up a batch job containing JCL and GBAT control records. There are two Geosupport interactive utility programs, both of them CICS transactions. The principal interactive utility is called the Geosupport Online Address Translator (GOAT), which provides general inquiry capabilities for almost all Geosupport functions. There is also a specialized interactive utility called Address / Intersection Map Zones (AIMZ), which displays a set of map identification numbers corresponding to an input address or intersection.

I.2 System Functionality

The output information that Geosupport provides consists of geographic information only. Geosupport does not provide, for example, population or crime statistics, housing data, building code violations, property ownership etc. Such data are available from the U.S. Bureau of the Census, city agencies and other sources. Geosupport can facilitate matching many such statistical and administrative data bases with user data containing individual locations, by associating those locations with district identifiers needed for such matching, such as census tract, zip code or tax block.

Geosupport processes New York City geography only, and is highly customized for that geography. For example, Geosupport can recognize and process many alternative names, spelling variants and partial names of New York City streets; the various address number formats that occur in the city; both old and new addresses on streets that have been renumbered; unique addressing schemes that exist in certain neighborhoods; and many other idiosyncrasies of New York City's geography.

Geosupport is organized into more than a dozen distinct functions that can be accessed by the user. Section I.4 contains a brief overview of Geosupport's suite of functions. The typical function accepts as input geographic locations of a particular type, such as addresses, street intersections or tax lots, and provides some or all of the following services, depending on the function and on calling options chosen:

- Geosupport standardizes and encodes components of the user input data. Specifically, it reformats input street names and input address numbers into standard formats, a process called 'normalizing', and it provides numeric street codes corresponding to input street names.

- Geosupport validates the input data. The nature of the validation performed depends on the function requested and the type of call made. Validation of geographic data is a particularly powerful tool in the interactive environment, where it can help applications to trap keying errors and street name misspellings at the point of initial data entry when such errors are most easily rectified.

- Geosupport geocodes the input data. That is, it outputs a predefined set of 'higher-level' geographic information associated with the input location, such as the community district, zip code, police precinct, cross streets.

- Geosupport enables consistent retrieval of user application data by geographic location. That is, it supports the ability of user applications to search (for inquiry or updating) or match their own data files by geographic location in a way that is independent of possible variations in referring to locations.

The nature of each application determines the combination of these services that is relevant. For example, some applications need only to validate geographic locations, not to obtain any of the output information that Geosupport provides. The fourth type of service, support for geographic retrieval consistency, is relevant only for applications that retrieve or match data from their own files by geographic location (as distinct from Geosupport's retrieval of data from its internal files). For those applications, geographic retrieval consistency is a critical issue. The next section contains a discussion of geographic retrieval consistency in general terms. Later chapters of this document contain detailed discussions of this topic.

I.3 Introduction to Geographic Retrieval Consistency

In applications that retrieve data from an application file by geographic location or match two application files by geographic location, the consistency of that retrieval or matching is a critical consideration that arises when processing any type of location that can be specified in more than one way. For example, consistency is a consideration for any type of location involving streets (such as addresses, intersections and street segments), since many streets have alternative names and many street names have spelling variants. The goal is to enable applications to retrieve records independently of which street name spelling was used when the record was created and which one is used at retrieval time. Similarly, consistency is a consideration when retrieving building-level data by address, since many buildings have more than one address. It is a consideration when retrieving data for street intersections, since many intersections (e.g., three-way intersections) can be specified using more than one pair of streets.

The achievement of retrieval consistency can greatly improve an application's 'hit' rate on geographic searches into the application's own files. Moreover, it enables applications to identify and consolidate multiple records for the same location effectively. These advantages can have a significant impact on the efficiency of a city operation. For example, an application can use this capability to generate a single work order for dispatching personnel to handle multiple repairs, inspections or other transactions for the same location.

Of the services that Geosupport provides, its use to achieve geographic retrieval consistency involves the most extensive integration of Geosupport in the design of the user application. Geosupport provides such support by returning certain data items which an application can store in its file during record creation and use as part of a geographic retrieval key. An example is an item called the five-digit street code, which applications can use to achieve consistent retrieval of data by those types of geographic locations that are specified in terms of streets. This is briefly discussed below, and is explained in detail in later chapters.

Within Geosupport, a set of numeric street codes has been assigned to represent New York City's street names. A full street code is a ten-digit item that, together with a borough identifier, corresponds to a specific spelling of a specific name for a specific street in that borough. The first five digits of the ten-digit street code are collectively called the five-digit street code. Ten-digit street codes are assigned in such a way that alternative names and spelling variants of the same street have the same five-digit street code. As a result, applications can achieve consistent retrieval or matching of application data by any type of geographic location that involves streets by using five-digit street codes instead of street names as part of the retrieval key. For the convenience of users, for all functions that involve street input except Function 1N and the display functions (Functions D, DG and DN), applications have the option to provide input streets to Geosupport in the form of either street names or street codes. Street codes are discussed in greater detail in Chapter IV.

I.4 The Geosupport Function Suite

This section contains a brief introduction to the Geosupport function suite. Each Geosupport function is identified by a one- or two-character function code. The function suite consists of location-processing functions (Functions 1, 1A, 1E, 2, 3, 3C, 3S, BL and BN), display functions (Functions D, DG and DN) and miscellaneous functions (Functions 1N, BB and BF). Table I-2 lists the currently implemented functions.

Each location-processing function processes input geographic locations of a particular type. For each type of location, there is an appropriate set of data items that collectively define such locations. Table I-1 lists the various types of geographic locations, the data items required to specify them, and examples. The location-processing functions can be sub-classified into address-processing functions, street-configuration-processing functions and ID-processing functions:

- The address-processing functions are Functions 1, 1A and 1E. They process conventional addresses and Non-Addressable Place names (NAPs).

- The street-configuration-processing functions are Functions 2, 3, 3C and 3S. They process geographic locations that are defined in terms of one, two or three streets, such as street intersections, block faces and street stretches.

- The ID-processing functions are Functions BL and BN. They process locations defined in terms of identifying numbers, namely, tax lot identifiers in the case of Function BL, and Building Identification Numbers (BINs) in the case of Function BN. Tax lot identifiers and BINs are discussed in detail in Chapter VI.

The address-processing functions differ from each other with respect to the output data that they provide and the nature of the validation processing that they perform. In general, the type of validation processing a Geosupport function performs is related to the geographic level of the output data. Thus, Function 1 validates only whether the input address falls within an address range for an entire block face, but it does not validate whether the input address is itself specifically valid. Function 1A, on the other hand, does validate whether the input address is a valid address for a specific building.

The display functions do not themselves directly ‘display’ anything, but they provide street names and/or address numbers in formats suitable for applications to display on screens, reports, mailing labels etc.

In Table I-1, the word ‘street’ refers to either a street name or a street code. In the examples in Table I-1, street names rather than street codes are used. (Note: the examples are formatted for reader comprehension, and would not be accepted by Geosupport as shown. Specifically, they contain borough names rather than the borough codes that Geosupport requires, and they contain English words and phrases such as ‘intersection of’ and ‘between’ that Geosupport does not recognize.)

Table I-1: Types of Geographic Locations Processed

Type of Location	Input Items Required to Specify Location, *Example*
Address	Borough + address number + street: *Bronx, 307 East Tremont Avenue*
Non-Addressable Place Name	Borough + place name: *Manhattan, Carnegie Hall*
Street Intersection	Borough + two intersecting streets: *Brooklyn, intersection of Flatbush Avenue and Atlantic Avenue* OR (if a pair of streets has two points of intersection), Borough + two intersecting streets + compass direction: *Queens, east intersection of Alderton Street and Cromwell Crescent* OR Borough + Intersection Name: *Manhattan, Isaac Stern Place*
Street Segment	Borough + 'on' street + two consecutive cross streets: *Manhattan, Broadway between W 38th St and W 39th St*
Block Face	Borough + street segment + compass direction specifying side of street: *Manhattan, east side of Broadway between W 38th St and W 39th St*
Street Stretch	Borough + 'on' street + any two cross streets: *Manhattan, Broadway between W 38th St and W 54th St* OR (if either or both of the cross streets has two points of intersection with the 'on' street), Borough + 'on' street + two intersecting streets + compass direction(s): *Queens, Alderton Street between east intersection with Cromwell Crescent and intersection with 63rd Drive* OR Borough + 'on' street: *Manhattan, Broadway*
Tax Lot	Borough + tax block + tax lot: *Staten Island, Block 247 Lot 16*
Building	Building Identification Number (BIN) *5006708*

Table I-2 below lists all of the current Geosupport functions, indicating for each function the type of input geographic location processed, the geographic level of the output data, and a sample of output data items. The table does not include normalized street names, street codes and normalized address numbers among the sample output items listed; those items are always returned when the input involves street names and address numbers. Certain terms not defined until later have been included in Table I-2 for completeness.

Table I-2: List of Geosupport Functions

Function	Type of Input	Description of Output Data	Sample Output Items
1	Address or Non-Addressable Place Name	Block face-related data	Cross streets, zip code, census tract and block, community district, police precinct, school district, health area, spatial coordinates (State Plane System)
1A	Address or Non-Addressable Place Name	Property-related data	Tax block and lot identifiers, list of all buildings, addresses and street frontages of property, condo flag, spatial coordinates
1E	Address or Non-Addressable Place Name	Block face-related data	Same as Function 1 + political districts: Election, State Assembly and Senate, City Council, Congressional and Municipal Court Districts
1N	Street Name or Place Name	Normalized name, street code	
2	Street Intersection	Intersection-related data	Additional streets at intersection (other than input streets), census tract, community district, spatial coordinates
3	Street Segment	Segment-related data + data related to left and right block faces	Cross streets, left and right zip code, left and right census tract and block, left and right community district
3C	Block Face	Block face-related data	Cross streets, zip code, census tract and block, community district
3S	Street Stretch	Street stretch-related data	Number of and list of intersections in order along the stretch, approximate distance in feet between intersections
BB, BF	Character String	See right-hand column	Set of ten normalized street names in alphabetical order.
BL	Tax Lot	Property-related data	Same as Function 1A
BN	Building	Property- and building-related data	Tax block and lot identifiers, list of all addresses of building, condo flag, spatial coordinates

Table I-2: List of Geosupport Functions (continued)

Function	Type of Input	Description of Output Data	Sample Output Items
D	5-Digit Street Code	Normalized 'primary' name of street	
DG	7-Digit Street Code	Normalized 'principal' name of local group	
DN	10-Digit Street Code	Normalized street name	

As a mnemonic aid, Geosupport function codes have been chosen to be as descriptive as possible. For functions involving street input, the first character of the function code is numeric and indicates the number of input streets. Thus, Functions 1, 1A and 1E process addresses and non-addressable place names, which are specified by a single input street or place; Function 2 processes intersections, which generally are specified by two input streets (although there are a few intersections that can be specified by a single intersection name); Functions 3, 3C and 3S process street segments, block faces and street stretches, respectively, all of which involve three input streets (an 'on' street and two cross streets), or, optionally, just an 'on ' street for Function 3S. The second character of the function code, if any, is descriptive as well: the letter 'C' signifies that the function involves compass direction input; the letter 'S' signifies street stretch input. The function codes of functions that do not involve street input are abbreviations of descriptive terms for the functions: BB and BF are abbreviations for 'browse backward' and 'browse forward', BL for 'block/lot', BN for 'building number', and D, DG and DN for 'display', 'display group' and 'display name'.

I.5 Overview of System Architecture

The Geosupport System consists of two major components called the foreground component and the background component, as well as the utility programs GBAT, GOAT and AIMZ. The relationships among the foreground component, the background component and the user application program are described in this section and are illustrated in Figure I-1 below.

Both the foreground component and the background component consist of both software and files. Users access the foreground component either directly from user-written programs via Geosupport API calls, or indirectly via the utility programs, which in turn access the foreground component via the Geosupport API. The foreground component and the utility programs are installed on IBM mainframes at the city computer centers listed in Appendix 7. Users do not access the background component, and it is not described in this document beyond the brief remarks in this section.

The Foreground Component

The files of the foreground component contain the geographic data that the foreground software requires to process user requests. User programs never read the foreground files directly; they are read only by the Geosupport foreground software.

The foreground software processes the input data passed to it by a calling user program. It performs such tasks as standardizing input street names and house numbers, reading foreground files, and returning information retrieved from those files, or appropriate error codes and messages, to the user program.

The Background Component

The background component contains a set of interrelated base files of the city's geography. The background files are continually updated and validated by the GSS staff. The background software includes software for updating and validating the background files and software for generating new foreground files from the background files.

The background work takes place partly on an IBM mainframe at the centralized data center operated by the City of New York Department of Information Technology and Telecommunications (DoITT), and partly in a specialized Geographic Information System (GIS) software environment running on Department of City Planning computers, where interactive computer mapping technology is used as a tool for updating some of the background files.

The background component, including GSS's GIS environment, is not accessible to users. To a user application, Geosupport appears to consist only of conventional data processing technology, and does not appear to include computer mapping capabilities. However, Geosupport, through its geocoding functionality, particularly its provision of spatial coordinates for an address, tax lot, or intersection, can facilitate the use of separate computer mapping or GIS software to display geographically-related user data graphically.

Foreground Component Updating: New File Releases

All of the foreground files are read-only files. They remain in production, unchanged, until GSS replaces them with a new set containing updated data. The set of foreground files in production at a particular time constitutes a release, and is identified by a release designator such as Release 02B. The first two characters of the release designator are the last two digits of the calendar year in which the release was deployed.

In the background component, GSS periodically performs a complex series of steps, called the Geosupport production cycle, to generate a new set of foreground files, quality assure those files, and deploy them for user access as a new Geosupport release.

Each new release is first implemented for user access on the DoITT mainframe. This is done in coordination with DoITT staff, who play an active role in migrating the files of the new release to all user-accessible CICS regions and the batch environment. After the new release is in production at DoITT for a brief trial period, GSS staff disseminate the new release to the other computer centers where Geosupport is installed.

For many applications, no special user action is required when a new release of Geosupport files is implemented; the application will continue to run as before. Of course, under the new release, Geosupport may respond differently to a particular set of input data than it had under previous releases. For example, it may return different output information for a given set of input data, it may accept input data that had previously been rejected, and it may reject input data that had previously been accepted.

In some applications in which data items obtained from Geosupport are stored in an application file, it may be appropriate for the user to update those stored items to reflect changes in each new Geosupport release. This is referred to as resynchronizing the user file with respect to the new Geosupport release. Resynchronizing is particularly important for applications that use Geosupport-provided items, such as street codes, in geographic retrieval keys. For such applications, the user should develop a resynchronization procedure, and should run that procedure each time a new release of Geosupport is implemented. Resynchronization is discussed further in Chapter IV.

Foreground Component Updating: New Software Versions and Vestigial Features

From time to time, GSS makes changes to the foreground software, to enhance the system or correct errors. The foreground software in production is identified by a version number, such as Version 9.2 etc. (Note that the foreground software is identified as a version while the foreground files are identified as a release.) Typically, new foreground file releases and new foreground software versions are installed in production independently of one another, and therefore there is not a one-to-one correspondence between file releases and software versions. Occasionally, a file release and a software version are implemented simultaneously.

It is a fundamental policy of GSS to strive to minimize the impact of Geosupport enhancements on existing applications. Whenever possible, enhancements are designed so that existing applications that do not require the new Geosupport feature need not be modified. In other words, enhancements are generally 'transparent' to existing applications. Although this is generally the policy of GSS, please see the **SUMMARY OF CHANGES AND NEW FEATURES**, at the beginning of this manual, for any item that could possibly affect your applications.

Over the years, numerous enhancements have been made to Geosupport, and virtually none of them have required existing applications to be modified or recompiled (except as necessary to take advantage of new features). As a consequence of this approach, Geosupport has a number of vestigial features. These are elements of the system, such as data items, work area formats, batch JCL, or entire functions, that are still operational but are obsolete or have been superseded.

Vestigial features will continue to be supported for the most part, so that existing applications that use them will continue to run without modification. However, vestigial features will not be enhanced. Moreover, vestigial features have that status because of some shortcoming. **Users are strongly encouraged to update their existing applications to eliminate all usage of vestigial features. All new applications should be designed to avoid any usage of vestigial features.**

Vestigial features are mentioned in appropriate sections of this document, and are identified as such, but in many cases they are not documented in detail. An example of a vestigial feature is the erstwhile Function 2C (superseded by an enhancement to Function 2; discussed in Section VII.2).

Character-Only Work Areas are an enhancement to Geosupport that was announced in Technical Bulletins in 2002. The Character-Only Work Areas are discussed, specifically, in Appendices 12, 13 and 14, and, in general, throughout the entire document.

User Feedback of Rejects

Typically, some of the geographic locations passed to Geosupport by a user application will be rejected as invalid. A reject could be caused by invalid user input data, such as a misspelled street name or an invalid address; or it could be caused by a Geosupport problem, such as an error or omission in Geosupport's internal data. Users should examine their rejects, and should report those rejects that cannot be attributed to user-caused errors to GSS staff, using procedures described in Appendix 6. In addition, users should report cases where Geosupport has accepted the input data but has returned output information that the user believes to be incorrect (for example, a zipcode that is believed to be incorrect for a particular input address). GSS relies on feedback from users as an essential source of information for quality-assuring Geosupport's data and keeping the data up-to-date and accurate.

GSS researches feedback received from users and updates the Geosupport background files as appropriate.

Such corrections become visible to user applications only after a new release of the foreground files reflecting the corrections is deployed for user access. A time lag of as much as several months is possible between the reporting of a reject to GSS and the appearance of the correction in the foreground component.

Figure I-1 below illustrates the basic architecture of the Geosupport System.

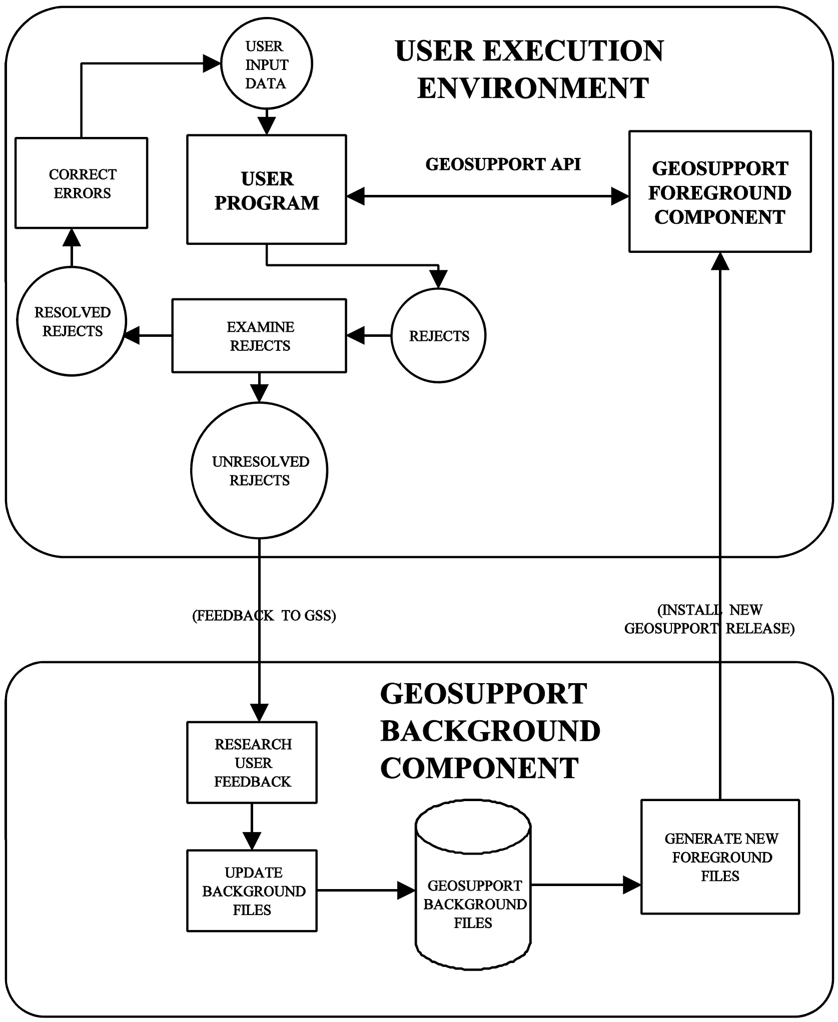

Figure I-1 Geosupport Architecture: Components and Production Cycle

Please note the following elements depicted in Figure I-1:

- The interaction between the application program and the Geosupport foreground component via the Geosupport API
- The examination by the user of rejects and the feedback of unresolved rejects to GSS for research and possible background file updating

- The periodic generation (in the Background Component) of new releases of foreground files

I.6 Overview of the User Programming Guide

This document, the *Geosupport System User Programming Guide* (UPG), contains the detailed technical information necessary for users to design and develop their own application programs that access Geosupport, as well as to use GBAT. (The use of GOAT and AIMZ requires no programming skills and they are not documented herein.) The UPG serves as a comprehensive set of technical specifications for the Geosupport System, and can be incorporated into procurement documents issued by city agencies soliciting consulting services for application design and development.

A separate publication, the *Geosupport System General Overview: Concepts and Facilities*, is a non-technical introduction to Geosupport for a general audience. It describes the goals and capabilities of Geosupport in general terms. In addition to the written documentation, GSS staff are available for consultation, within the limits of the section's resources, during all phases of user application planning, design, development and implementation.

The contents of the UPG are as follows.

- This chapter, Chapter I, is a general overview of the system, its purposes, features, means of access and the broad outlines of its architecture.

- Chapter II is an introduction to the Geosupport API, describing in general terms its components and the user programming required to utilize it. (Chapter VIII discusses the usage of the API in greater detail.)

- Chapter III covers street name processing and in particular describes important user-controllable features of Geosupport's street name standardizing routine.

- Chapter IV discusses Geosupport's system of numeric street codes, a feature that is relevant principally for applications that must retrieve data from user files by geographic location.

- Chapters V through VII discuss in detail the types of geographic locations that Geosupport can process and the functions that process them:

    - Chapter V discusses address and non-addressable place name processing and Functions 1, 1A and 1E.

    - Chapter VI discusses tax lot and building processing and Functions 1A (aspects not covered in Chapter V), BL and BN.

    - Chapter VII discusses street configuration processing and Functions 2, 3, 3C and 3S.

- Chapter VIII describes in detail the application program coding and JCL required to use the Geosupport API.

- Chapter IX discusses GBAT, Geosupport's standalone batch utility program.

- Appendix 1 contains summary reference information for each Geosupport function, including a brief description of the function, the required and optional input items, the Geosupport return codes specific to the function, and references to relevant sections of the UPG.

- Appendix 2 contains layouts of the Geosupport API work areas.

- Appendix 3 is a data item dictionary, containing descriptions of the formats and contents of all of the data items that appear in the work areas.

- Appendix 4 is a comprehensive table of Geosupport Return Codes, Reason Codes and Messages.

- Appendix 5 contains listings of the MSW Geosupport COPY files for all of the programming languages supported by the Geosupport COPY feature. (This important feature is discussed in Chapter VIII.)

- Appendix 6 describes the procedures that users should follow to report Geosupport System problems and to provide feedback to GSS of rejected input data that the user is unable to resolve.

- Appendix 7 is a list of the data centers where Geosupport is installed.

- Appendix 8 contains sample user programs written in various programming languages together with sample JCL.

- Appendix 9 contains reference tables for setting up GBAT jobs.

- Appendix 10 contains sample GBAT jobs.

- Appendix 11 contains a set of guidelines for user application design.

- Appendix 12 contains a description of Character-Only Work Areas (COWs) and how to use them.

- Appendix 13 contains layouts of the Character-Only Work Areas.

- Appendix 14 contains listings of the COW Geosupport COPY files for all of the programming languages supported by the Geosupport COPY feature. (This important feature is discussed in Chapter VIII.)

- The Glossary contains definitions of special terms and acronyms.

Appendices 1 through 5 collectively can serve as a quick reference guide for experienced Geosupport users.

Note: Since the geography of New York City is constantly growing and changing, any examples mentioned in this document may, occasionally, function differently from the way they are described. The examples will, in any case, illustrate the concept being discussed.

CHAPTER II: INTRODUCTION TO THE GEOSUPPORT API

II.1 Introduction

This chapter presents an overview of the Geosupport Application Programming Interface (API), the mechanism through which a user-written application program interfaces directly with the Geosupport System. The basic architecture of the API, the user programming procedure required to utilize the API, and reject handling are described. The important distinction between one-work-area and two-work-area calls is discussed, and the long-work-area-2 option is described. The material in this chapter is general in nature. Chapter VIII discusses in detail the user programming statements and JCL required to utilize the API, and other chapters discuss application design issues specific to the various functions.

The Geosupport API consists of the following elements:

- A Geosupport load module called the driver that the user must link-edit into the application program. The driver serves as an intermediary between the user's application program and the Geosupport foreground software.

- One or two standard-layout work areas that the user must include in the application program and that are used to pass data between the application program and Geosupport.

- Programming statements that the user codes in the application program utilizing the driver and work area(s) to interact with Geosupport.

With very few exceptions, the Geosupport API is identical in the batch and CICS environments. The principal exception is the name of the driver.

The driver has two principal purposes. It passes execution control from the user program to the Geosupport foreground software, which is external to the user program load module. The driver also passes addressability to the work areas (which are located within the user program) to the Geosupport foreground software, thereby enabling the foreground software to access those work areas.

User programs never read Geosupport's internal files directly. They are read only by the Geosupport foreground software, which returns data retrieved from those files to the calling user program in the work areas. In batch applications, the user JCL for the execute step must include DD statements for the load libraries that contain the Geosupport foreground software and data. Chapter VIII describes the JCL required for batch execution, and Appendix 8 contains examples.

Figure II-1, below, illustrates the elements of the Geosupport API as just described. The illustration assumes that the user program has a data file into which it writes information obtained from successful calls to Geosupport, and another file, printed report or screen display for handling rejects.

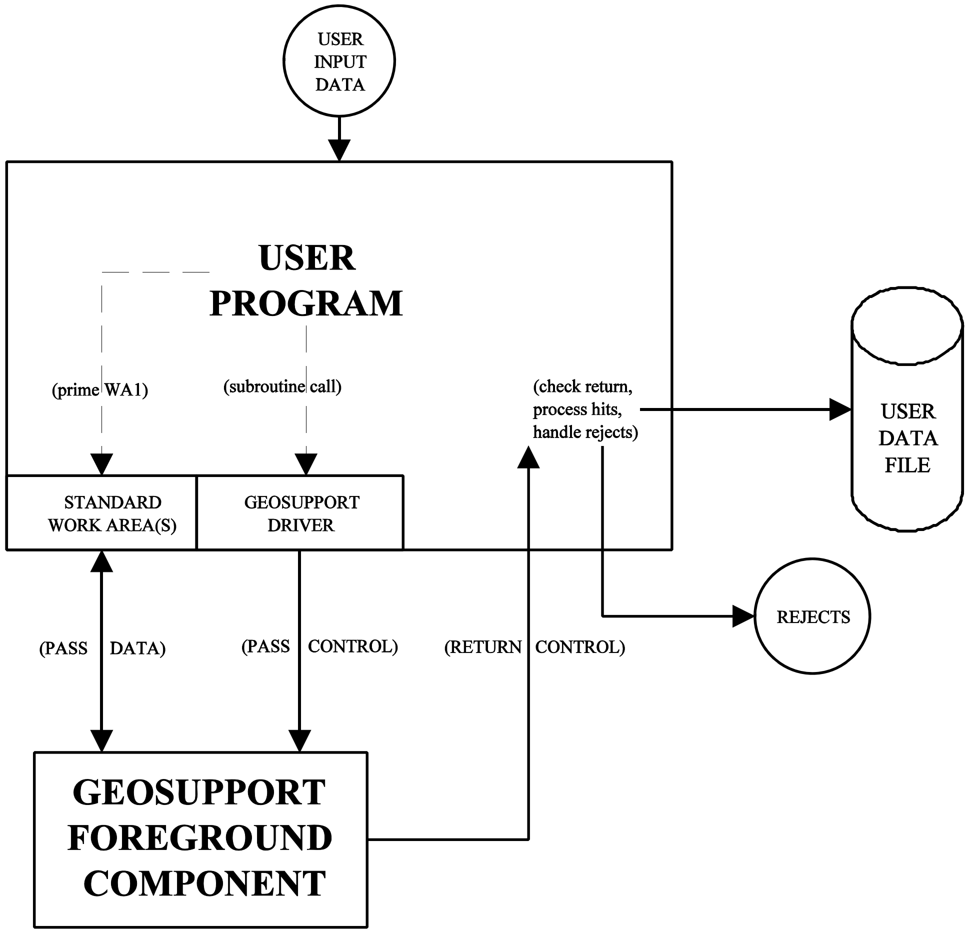

Figure II-1: The Geosupport API

When a user program issues a call to the driver, either one or two work areas are passed as parameters of the call. Work Area 1 (WA1) is always passed, and its length and layout are the same for all functions. Work Area 2 (WA2) may also be passed, depending on the Geosupport function being requested and the type of information needed by the user's application. The length and layout of WA2 are determined by the function and, for functions that have the 'long Work Area 2 option', by whether that option is specified. The distinction between one-work-area calls and two-work-area calls is discussed in Section II.4. The long Work Area 2 option is discussed in Section II.5. The work areas may be Mainframe-Specific Work Areas (MSWs) or Character-Only Work areas (COWs). For a description of these formats see Appendix 12.

For the convenience of users whose programs are written in COBOL, IBM mainframe assembler, PL/1, C or NATURAL, Geosupport COPY libraries are maintained, containing source code layouts of each work area in each of those programming languages. The Geosupport COPY libraries are discussed in detail in Chapter VIII. The use of the Geosupport COPY libraries is optional but strongly recommended.

Note for CICS NATURAL Users: In order for Geosupport's CICS driver to pass control to the foreground component of Geosupport properly, it must know whether the calling user program is written in NATURAL. The driver determines this by examining a Geosupport table. This table contains a list of the transaction-IDs of CICS transactions that launch NATURAL programs that call Geosupport. If the transaction-ID of such a transaction is not in the table, the driver will mistakenly assume that the program calling Geosupport is not written in NATURAL, and the transaction will terminate abnormally when the program calls Geosupport.

At DoITT, the updating of the Geosupport NATURAL transaction-ID table is the responsibility of DoITT staff. Therefore, DoITT users who have new CICS applications written in NATURAL that are to access Geosupport must so inform DoITT staff, who will enter the new transaction-ID into the table. CICS NATURAL users running at other computer centers should contact GSS.

II.2 Geosupport Return Codes and Reject Handling

Geosupport has an elaborate apparatus to support application problem-handling. There are three output fields in Work Area 1 that are used to inform calling applications of the outcome of each call to Geosupport. These fields are the Geosupport Return Code (GRC), the Reason Code and the Message. A comprehensive list of GRCs, Reason Codes and Messages is contained in Appendix 4.

The GRC is a two-byte character item into which Geosupport inserts a value before returning control to the calling application, as follows.

- A GRC value of '00' indicates unconditionally successful completion.
- A GRC value of '01' indicates a warning condition.
- A GRC value other than '00' or '01' signifies unsuccessful completion, or rejection, caused by either a system error or a user error.

Warnings are conditionally successful completions. They alert the user to unusual aspects of the input or output data, or signify that Geosupport made an assumption about or modification to the input data. For example, Function 1 issues a warning to alert the user that a required hyphen is missing from an input house number and that Geosupport has inserted the missing hyphen into the output normalized house number.

It is advisable for application designers to review the possible warnings that can be elicited by the functions their application will be calling (delineated in Appendix 4), and to determine whether there are types of warnings for which it would be appropriate to provide special handling routines. In some applications, it may be appropriate simply to display the messages that accompany warning returns, and otherwise to process warnings in the same fashion as unconditionally successful completions.

For all unconditionally successful completions, Geosupport returns values in the work area(s) for the full set of output data items that the given Geosupport function is designed to provide. (But see the note of caution regarding the return of values in work area output fields in Section II.4.) In the case of a warning, certain output fields may be 'empty' (blank, all zeros or otherwise devoid of information), depending on the nature of the warning. In the case of a rejection, almost all output fields are returned empty, but there will be values in the GRC and Message.

Rejects can be caused either by a system error or a user error.

- System errors are problems that are not attributable to the user program or to the user input data, and therefore are beyond the user's control. Typical system errors are hardware errors, operating system errors and Geosupport software errors.

- User errors can occur when the user's program makes a call to Geosupport improperly, such as passing an invalid number of work areas; or when Geosupport considers the user's input data to be geographically or otherwise invalid, such as an invalid function code, an empty input field for which a value is mandatory, an invalid borough code, an invalid address.

For all warnings and rejects, the Message field contains an appropriate message. In addition, for all warnings and for some rejects, a value is returned in the Reason Code field, specifying more precisely the reason for the warning or rejection.

The user program should be designed so that, immediately upon receiving execution control back after a call to Geosupport, it examines the GRC (and the Reason Code, when relevant) to determine the outcome of the call, and takes appropriate action. (Note: the textual content of Geosupport messages is subject to revision without notice. Therefore, application developers should program rejection processing based on the value of the GRC and Reason Code rather than on the Message.) In batch applications, appropriate actions for processing a warning or reject might include printing out the GRC, Reason Code and Message and/or writing the record to a reject file. In interactive applications, appropriate actions might include displaying the GRC, Reason Code and Message on the screen, and (for user errors, not system errors) giving the data entry operator an opportunity to correct the error and resubmit.

II.3 Geosupport API User Programming Procedure

A field in a Geosupport API work area into which the user program inserts a value to be passed to Geosupport is referred to as an input field of the work area (because it is an input datum to Geosupport). A field in a work area into which Geosupport inserts a value to be returned to the user program is called an output field. WA1 contains both input and output fields. WA2 contains output fields only.

The loading of values into WA1 input fields by the user program prior to issuing the call to the driver is referred to as priming WA1. The function being requested, determines which WA1 input fields must be primed, which are optional, and which are not used. One WA1 input field that is mandatory for all calls is the field for the function code. Combinations of other WA1 input fields, such as those for borough code, address number, street name and street code fields, collectively serve to specify a geographic location to be processed. Still other WA1 input fields are for specifying processing options, such as parameters that control how street names are normalized; most of those fields have default values and are optional.

It is essential that the user program clear WA1 to blanks before priming it, in order to eliminate any 'stray' data inadvertently lingering from a previous call. Various fields in the WA1 output area are initialized to blanks to help avoid extraneous data. If the call involves two work areas, however, WA2 need not be cleared by the user program before calling the driver, because Geosupport clears WA2 automatically.

After clearing WA1 to blanks, the user program primes WA1, and then issues a standard subroutine call to the driver, passing the work area(s) (more precisely, their memory addresses) as parameters of the call. (Note that a standard subroutine call is used to call the driver even in the CICS environment, rather than a CICS LINK.) The driver, in turn, passes execution control to Geosupport (more precisely, to the foreground component of Geosupport), which is external to the user program load module.

When Geosupport completes its processing for the given call, control is returned to the driver, which in turn returns control back to the user program. The user program can issue any number of calls to Geosupport during a single execution. Each call is an independent event, which Geosupport processes based entirely on the contents of WA1 passed in that call; Geosupport does not 'remember' previous calls.

The procedure that a user program would follow to call Geosupport via the API can be summarized as follows:

1) Clear WA1 to blanks.

2) Prime WA1. That is, move values to the appropriate input fields in WA1. The function code is always required; other required and optional input fields depend on the function, and are listed in Appendix 1.

3) Issue a standard subroutine call to the driver, passing as calling parameter(s) either WA1 only or both WA1 and WA2. The required calling statements are described in Sections VIII.3 and VIII.5. The distinction between one- and two-work-area calls is discussed in Section II.4.

4) Upon return of control to the user program, examine the GRC (and the Reason Code, if appropriate) in WA1, and take appropriate action. A list of the GRCs and Reason Codes that can be produced by each function is in Appendix 4.

II.4 One-Work-Area and Two-Work-Area Calls

There are important distinctions between one-work-area and two-work-area calls. When a Geosupport function is called using one work area, Geosupport 'normalizes' certain input items, that is, it reformats them into a standard form. For each such input item that Geosupport successfully normalizes, there is a WA1 output field into which Geosupport inserts the item in normalized form. Normalizing includes such processing as right-justifying and zero-filling certain numeric input items (such as tax block and tax lot numbers), and providing fully spelled out borough names corresponding to input borough codes. Normalizing also encompasses performing complex algorithms to reformat street names and address numbers into standard formats.

After normalizing the appropriate input items, if the one-work-area call involves street name input items, Geosupport attempts to retrieve the street code corresponding to each input street name. If this is successful, each street code is returned in WA1. In summary, the processing Geosupport performs for a one-work-area call consists of normalization of the input data and the return of normalized values and street codes (if any) in WA1.

The processing performed for a two-work-area call includes all of the processing performed for a one-work-area call as well as certain additional processing. The nature of this additional processing depends on the Geosupport function. The additional processing generally consists of accessing Geosupport files in order to attempt to obtain certain geographic information associated with the input data. If the file access is successful, this geographic information is returned in WA2 (but see the cautionary note below). The process of associating higher level geographic information with an individual location is called <u>geocoding</u>, and the items of higher level information are called <u>geocodes</u>. Typical examples of geocodes returned by Geosupport in WA2 are community district, census tract, zip code and health area.

Caution: For a two-work-area call, a GRC of '00' or '01' signifies that Geosupport has accepted as valid the input geographic location specified by the user, but it does not guarantee that every item normally returned by the given function in WA2 contains a non-empty value. If a WA2 field is returned empty, this may or may not be erroneous. The field might be empty (blanks, zeros or otherwise devoid of information) because of an erroneous Geosupport data omission; this should be reported to GSS staff using the feedback procedures described in Appendix 6. However, the field might be empty intentionally and non-erroneously because the type of geographic area it represents does not completely cover the city. For example, there are certain non-residential areas of the city where the Department of Sanitation has not defined Collection Scheduling districts. When a two-work-area call results in a GRC of '00' or '01', it is the responsibility of the user program to determine whether the particular WA2 fields being used by the application are non-empty.

Except for system errors, the outcome of a call to Geosupport, as signified by the GRC, Reason Code and Message, concomitantly has significance with respect to the geographic validity of the input data. The type of validation performed depends on the function and on whether a one- or two-work-area call has been made. The validations performed in a two-work-area call to a function are always more extensive than those performed in a one-work-area call to the same function. To illustrate this, consider Function 2, which processes an intersection specified in terms of two streets. In a one-work-area Function 2 call, if the two input streets were specified in the form of street names rather than street codes, Geosupport attempts to normalize the street names and obtain their street codes; success in doing so *ipso facto* validates that each input street name is recognizable to Geosupport as a valid name of a specific New York City street. However, the two input streets (names or codes) are processed independently of each other and are not treated as collectively defining a geographic location, in this case an intersection. In other words, when Function 2 is called using one work area, the existence of the intersection formed by the two input streets is not validated; the call will result in a GRC of '00' or '01' if both input street names are successfully normalized and recognized, regardless of whether the two streets intersect. In a two-work-area Function 2 call, on the other hand, Geosupport treats the two input streets as the intended specification of an intersection; an (unconditionally or conditionally) successful outcome *ipso facto* validates the existence of this intersection, and if it is valid, Geosupport returns information about the intersection in WA2. For Geosupport functions in general, a one-work-area call validates only that the input items can be normalized and that input street names are recognizable to Geosupport, while a two-work-area call additionally provides some level of validation of the geographic location specified collectively by the input items. The specific validations performed in a two-work-area call to each Geosupport function are described in Chapters V through VII.

A two-work-area call causes Geosupport to access files additional to those accessed for a one-work-area call to the same function. Therefore, to maximize execution efficiency, when an application does not require the additional output data and/or validation processing that a two-work-area call provides, the application should issue a one-work-area call.

II.5 The Long Work-Area-2 Option

From time to time, GSS adds new output fields to a function's WA2. For example, in 2001, the WA2s of several functions were enhanced to include fields for the 2000 census tract and block. In general, when new output fields are added to a Geosupport work area, GSS utilizes existing filler space in the work area for those fields, if available. In that way, the basic layout of the work area remains the same, and existing users of that function who do not need to make use of the new items are not compelled to modify their applications.

If there is not enough filler space available in a function's WA2 to accommodate new fields, GSS introduces a 'long WA2 option' for that function, as described below. This approach enables new data items to be made available to applications that need them, without affecting existing applications that do not need them. The MSW functions that currently have the long WA2 option are Functions 1, 1E, 1A, BL and 3. The COW functions that currently have the long WA2 option are functions 1A and BL. The long WA2 option may be implemented for other functions in the future.

When issuing a two-work-area call to a function that has the long WA2 option, the application has the option to use either the 'regular WA2' (the work area layout that had already been in existence before the long WA2 option was introduced for that function), or the 'long WA2'. The application informs Geosupport that the long WA2 is being used by inserting an 'L' in a WA1 input field called the Long Work Area 2 Flag. When the long WA2 option is specified, it is the application's responsibility to pass a WA2 of the proper length to the Geosupport driver. If the application passes a blank in the Long WA2 Flag, the regular WA2 is used. Both the regular and long WA2s are documented in Appendix 2.

The MSW Function 3 exemplifies the role of the long WA2 option. MSW Function 3's regular WA2 is 200 bytes long, almost all of which was long ago allocated to specific fields, leaving little filler space available for new fields. At some point in the past, the necessity to add new fields for which no space was available in Function 3's regular WA2 impelled the introduction of the long WA2 option for Function 3. Function 3's long WA2 is 300 bytes long, and consists of the same 200 bytes of information that are returned in the regular WA2, followed by 100 additional bytes containing fields for several additional items that the regular WA2 was not designed to include, as well as ample filler space for future enhancements. Applications that existed prior to the introduction of the long WA2 option for Function 3, and that have no need of any of the fields returned in the last 100 bytes of the long WA2, are able to continue running properly without modification using the regular WA2.

CHAPTER III: STREET NAME PROCESSING

<u>III.1 Introduction</u>

This chapter discusses Geosupport's street name processing in detail. (In this chapter, unless otherwise noted, the term 'street name' is used generically to encompass not only names of city streets, but also a wide variety of other New York City geographic feature names that Geosupport recognizes, including the names of some tunnels, bridges, rail lines, shorelines and geographic place names of various kinds.) The street name normalizing algorithm is briefly outlined. Two aspects of street name normalizing that are under user control, the selection of a street name normalization format and the Street Name Normalization Length Limit (SNL) parameter, are described. Other street name processing features that are described are partial street names, similar names, and street name browsing. Certain non-street feature names, place names and 'pseudo-street names' that are recognized by Geosupport are also discussed in this chapter. The related topic of street codes is discussed in detail in Chapter IV.

It is important to note that New York City geographic names are meaningful only when the borough is identified, since features in different boroughs can have the same name. For example, all five boroughs have a street named BROADWAY.

Applications pass up to three input streets to Geosupport in a single call, depending on the function being called. For most of the functions that accept street input, input streets are passed either in the form of street names or in the form of street codes. The exceptions are that Function 1N accepts street name input only, and the display functions, Functions D, DG and DN, accept street code input only.

Input streets are passed to Geosupport using as many as necessary of WA1's three input street name fields or its three input street code fields. Each WA1 input street name field is 32 bytes long. If there is more than one input street in a call, they must all be of the same type, either all names or all codes, not a combination of both types. If both street names and street codes are specified in WA1, for all functions other than D, DG, and DN, Geosupport processes the street names and ignores the street codes. For functions D, DG, and DN, the street names are ignored.

When street input is in the form of street names, before attempting to identify which New York City street an input name refers to, Geosupport attempts to 'normalize' the name by executing a systematic algorithm intended to produce a version of the name in a standardized format. If normalization is successful, Geosupport returns the normalized street name(s) to the user in as many as necessary of WA1's three output normalized street name fields. Geosupport's normalizing algorithm is designed so that users have considerable leeway in spelling input street names. For example, input names may contain commonly used abbreviations for words like avenue, street, boulevard, east, etc.

When Geosupport is able to normalize an input street name successfully, it uses the normalized name to read an internal Geosupport file in order to obtain the street code. Successful normalization followed by successful street code retrieval *ipso facto* constitutes Geosupport System validation of the input street name, i.e. its identification or 'recognition' as the name of a specific New York City street. Note that successful normalization alone does not constitute validation of the input street name.

Geosupport's street name normalizing algorithm is highly customized for New York City. The algorithm is complex and a complete description of it is beyond the scope of this document. In any event, the algorithm is performed automatically, and users need to be aware primarily of two aspects that they can control. These are a parameter for controlling the maximum length of normalized street names, called the SNL; and a choice of two formats for normalizing street names, called the compact and sort formats. These features are described in detail in this chapter. For completeness of the discussion, and because some familiarity with the normalizing algorithm may aid the user in understanding possible causes of rejection, a summary description of the normalizing algorithm is also given in this chapter.

Function 1N. Function 1N can be used to normalize a street name and retrieve its street code, without having to specify a particular geographic location. Function 1N requires the input only of a borough code and a street name. The SNL parameter and the selection of a street name normalization format can be specified in a Function 1N call. Function 1N is called using Work Area 1 only.

III.2 Street Name Normalizing and the SNL Parameter

Street name normalizing is governed by a user-controllable parameter called the Street Name Normalization Length Limit (SNL), which sets an upper limit to the lengths of output normalized street names. The SNL feature is particularly useful in applications that have a restricted amount of space for the display of street names, such as when addresses must be visible through transparent envelope windows, or when a screen display or printed report line is crowded.

The user specifies an SNL value using the two-byte WA1 input SNL field. The permissible range of SNL values is 4 through 32, inclusive. The setting of an SNL value is optional. If the user specifies no SNL value, the default value of 32 is in effect for that call to Geosupport. Every call to Geosupport is an independent event, even within a single execution of a user program, so if an SNL value other than 32 is desired in a particular call, it must be explicitly specified in that call; Geosupport does not 'remember' an SNL value specified in a previous call.

Geosupport attempts to normalize each input street name in such a way that the result has a length in bytes that does not exceed the SNL value in effect. The SNL also governs the length of the normalized street name output returned by the display functions (Functions D, DG and DN). However, the SNL does not limit the lengths of input street names. Regardless of the SNL value, the maximum length of an input street name is 32 bytes, which is the length of the WA1 street name input fields.

The smaller the SNL value that the user specifies is, the more difficult it is for Geosupport to normalize input street names within that length limit, and therefore the greater the proportion of input street names that are likely to be rejected as not normalizable. Consequently, users who must limit the lengths of normalized street names should specify the largest possible SNL value that can satisfy the needs of their application. An SNL value of 32 (the default) insures that virtually all New York City street names can be normalized. It is recommended that in the design of new applications, 32 bytes be allocated for street name fields in files, programs, screens, reports and manual forms whenever possible.

The following is a simplified description of the street name normalizing algorithm:

- Parsing the input name: The normalizing algorithm logically separates the input name into 'words' delimited by blanks. Any sequences of consecutive blanks are consolidated to single blanks. If any numeric characters (the digits '0' through '9') and non-numeric characters are adjacent to each other, they are separated by the insertion of blanks. For example, W2PLACE becomes W 2 PLACE.

- Deleting ordinal suffixes: Numeric words in street names are often expressed as ordinal numbers (integers formatted to specify order, consisting of numeric digits followed by ordinal suffixes, such as '1st', '2nd', '3rd', '4th'). The normalizing algorithm deletes the ordinal suffixes (the endings 'st', 'nd', ‘rd’ and ‘th’) from such words. For example, WEST 3RD STREET is converted to WEST 3 STREET. Note, however, that numeric words that are expressed alphabetically (such as WEST THIRD STREET) are not modified.

- Handling special characters: The normalizing algorithm deletes any periods (the character ‘.’) at the ends of words. For example, ST**.** MARKS PLACE becomes ST MARKS PLACE. Any periods not at the ends of words are replaced by blanks, which will usually cause rejection. Special characters other than periods are left unaltered, and will cause rejection unless those special character(s) are specifically valid for the given street name. (Currently, the only special characters that appear in specific street names accepted by Geosupport are apostrophes and hyphens.) In general, if Geosupport accepts a street name with a special character, it will also accept that street name without the special character. For example, in Manhattan, both SAINT MARK’S PLACE and SAINT MARKS PLACE are accepted. In the Bronx, O’BRIEN AVENUE, OBRIEN AVENUE and O BRIEN AVENUE are all accepted. In Manhattan, BEN-GURION PLACE, BEN GURION PLACE and BENGURION PLACE are all accepted.

- Expanding and abbreviating standard words under SNL constraint: There are certain standard words that appear frequently in street names, either fully spelled out, such as EAST, AVENUE and BOULEVARD, or in the form of standard abbreviations, such as E, AV or AVE, and BL or BLVD, respectively. If the input name is shorter than the SNL value in effect, then to the extent permitted by that SNL value, the normalizing algorithm expands standard abbreviations to their full spellings. Conversely, if the input name is longer than the SNL value in effect, then the normalizing algorithm attempts to shorten the name to the extent required by that SNL value, by replacing fully spelled out standard words with standard abbreviations.

- Suppressing expansion in special cases: The normalizing algorithm recognizes certain special cases in which a character string normally treated as a standard abbreviation is not to be so treated, that is, is not to be expanded under any circumstances. For example, ST is expanded to STREET only when it occurs as the last word of the input name; this prevents the conversion, for example, of ST MARKS PLACE into STREET MARKS PLACE. Certain character strings that are treated as standard abbreviations in most street names are not so treated in specific street names; for example, the ‘S’ in the Brooklyn street name AVENUE S and in the Bronx street name S STREET is not expanded into SOUTH; the ‘E’ in the Manhattan street name ABRAHAM E KAZAN STREET is not expanded into EAST; the ‘DR’ in the Manhattan street name DR MARTIN L KING JR BOULEVARD is not expanded into DRIVE.

III.3 Street Name Sorting and Normalization Format Options

Many applications display addresses or other types of geographic locations in their reports and online screens, including normalized street names obtained from Geosupport. Applications often sort their data by geographic location for display. However, street names that contain numeric characters do not sort appropriately when they have been normalized in the ‘conventional’ fashion. To solve this problem, Geosupport is able, at the user’s option, to normalize street names either into the conventional format, which is called the compact format, or into a format that is more suitable for sorting, called the sort format.
The compact and sort formats differ only for street names that contain numeric characters. Such a street name contains, in the sort format, a number of ‘alignment’ blanks in front of the numeric digits in the street name, which serve to align the numeric digits for proper sorting. In the compact format, no alignment blanks are present. The presence or absence of the alignment blanks is the sole difference between a name that contains numeric characters normalized in the sort format and the same name normalized in the compact format. We illustrate by displaying, side by side, two sorted lists of a sample of Manhattan street names

normalized in the two formats:

```
SORTED LIST IN COMPACT FORMAT         SORTED LIST IN SORT FORMAT

EAST HOUSTON STREET                      5 AVENUE
EAST 1 STREET                           10 AVENUE
EAST 10 STREET                        EAST    1 STREET
EAST 102 STREET                       EAST    2 STREET
EAST 129 STREET                       EAST    3 STREET
EAST 13 STREET                        EAST    9 STREET
EAST 167 STREET                       EAST   10 STREET
EAST 2 STREET                         EAST   13 STREET
EAST 20 STREET                        EAST   20 STREET
EAST 201 STREET                       EAST   79 STREET
EAST 3 STREET                         EAST  102 STREET
EAST 79 STREET                        EAST  129 STREET
EAST 9 STREET                         EAST  167 STREET
FULTON STREET                         EAST  201 STREET
10 AVENUE                             EAST HOUSTON STREET
5 AVENUE                              FULTON STREET
```

As this example illustrates, in the compact format, normalized street names do not sort appropriately. For example, EAST 10 STREET sorts in front of EAST 9 STREET, and 10 AVENUE sorts in front of 5 AVENUE. In contrast, in the sort format, the presence of the alignment blanks causes street names containing numeric characters to sort appropriately. Notice that the presence of the alignment blanks in the sort format, and their absence in the compact format, causes a change to the sort order of numeric street names not only relative to each other, but also relative to non-numeric street names. For example, in the compact format, FULTON STREET sorts in front of street names that begin with a numeric character, such as 10 AVENUE, while in the sort format it sorts behind them. Similarly, in the compact format, EAST HOUSTON STREET sorts in front of the street names that start with the word EAST followed by a numeric word, while in the sort format, it sorts behind those street names.

Note that for purposes of this discussion, all samples of sort output assume the EBCDIC collating sequence.

The sort format should always be used for street names that are to be sorted. However, the sort format is not as well-suited for display purposes as the compact format, since the alignment blanks give the sort format an awkward appearance. In applications that must display data sorted by geographic location, sorting should be done using street names in the sort format, while street names should be displayed in the compact format. (This would, of course, necessitate the application making a second call to Geosupport for each name, to obtain the alternative format. Function 1N could be used for that purpose.)

The sort format is the default format. That is, Geosupport will normalize input street names into the sort format unless the user program specifically requests the compact format by placing a ‘C’ in the Street Name Normalization Format Flag field in WA1. Note that every Geosupport API call is an independent event: Geosupport does not ‘remember’ previous calls. Therefore, if repeated calls are being made within a single execution of an application program, and the user wishes all the input street names to be normalized into the compact format, a ‘C’ must be present in the flag during each call.

We now give a precise description of the sort format. First, note that New York City street names have numeric characters (the digits ‘0’ through ‘9’) in at most one word. If a street name has such a ‘numeric word’, that word consists only of a one-, two- or three-digit number, possibly followed by an ordinal suffix.

(If there is an ordinal suffix, it is deleted during normalizing in either format.)

For street names that do not have a numeric word, the compact and sort formats are identical. For a street name that does have a numeric word, the two formats differ only in the fact that alignment blanks are present in the sort format and absent in the compact format. In forming the sort format, the normalizer inserts the required number of alignment blanks in front of the numeric characters, to form a four-byte field within which the numeric characters are right-justified and blank-filled. (The rationale for using four bytes for the normalized numeric word is explained below.) Thus, when normalizing street names that have a numeric word into the sort format, the normalizer inserts three blanks in front of a one-digit number, two blanks in front of a two-digit number and one blank in front of a three-digit number. The inserted alignment blanks are additional to the single word-separating blank between the numeric word and the preceding word in the street name, if any.

We illustrate with an example, using the dash character to represent blanks for clarity. The street name EAST--129 STREET is in sort format. The first blank between EAST and 129 (represented by the leftmost dash) is the word-separating blank always present (in either format) between any two consecutive words. The second blank is the alignment blank inserted only in the sort format to right-justify the three-digit number '129' within the four-byte field for the numeric word. EAST-129 STREET is the same street name in compact format; it has the single word-separating blank between the two words, but no blank inserted for alignment.

Note that the sort format is designed so that numeric words are normalized right-justified into a four-byte field, even though numeric words in New York City street names never have more than three digits. The purpose of the extra byte is to insure that non-numeric street names do not sort between street names with numeric words having fewer than three digits and those that have exactly three digits. The four-byte field assures this, since it causes the first position of the normalized numeric word always to be a blank.

The following example illustrates the advantage of using a four-byte field for normalizing the numeric word. Below we display two sorted lists of the same five street names. In the first list, the names have been normalized in Geosupport's actual sort format, that is, using a four-byte field for the numeric word. In the other list, they have been normalized in a hypothetical sort format, using a three-byte field. In both lists, alignment blanks are represented by dashes, and word-separating blanks are represented by spaces.

Actual Sort Format With 4-Byte Numeric Word Field	Hypothetical Sort Format With 3-Byte Numeric Word Field
`EAST ---7 STREET`	`EAST --7 STREET`
`EAST --23 STREET`	`EAST -23 STREET`
`EAST -129 STREET`	`EAST HOUSTON STREET`
`EAST -203 STREET`	`EAST 129 STREET`
`EAST HOUSTON STREET`	`EAST 203 STREET`

In this example, all of the street names are identical in their first five positions, with the fifth position being a word-separating blank. In the four-byte list, all the numeric names have a blank in the sixth position (the first position of the four-byte numeric field), and therefore have sorted ahead of the one non-numeric name, which has an 'H' in that position. In the three-byte list, the numeric names containing fewer than three digits have a blank in the sixth position, the non-numeric name has an 'H' there, and the numeric names containing three digits have a numeric character (a '1' or a '2') in the sixth position. Since the sort sequence of these characters is blank, 'H', '1', '2', the result of sorting with a three-byte numeric field is the undesirable separation of the numeric names by the non-numeric name.

III.4 Partial Street Names

It is a common informal practice to refer to streets using partial versions of 'full' street names. For example, the intersection of Nassau Street and Broad Street in Manhattan might be specified as the intersection of "Nassau and Broad". To accommodate this practice, Geosupport is designed to accept such partial street names as input street names whenever feasible. In this section, a precise definition and some examples of partial street names are given, and the circumstances under which Geosupport accepts a partial street name as an input street name are described.

A partial street name is a character string that is not itself a valid 'full' street name, and that is formed from a valid full street name by deleting one or more entire words from the end of the full street name. Note that, according to this definition, forming a partial street name involves the deletion of words only from the end of a full street name, not from the beginning or middle, and the deletion only of entire words, not portions of words. The following examples illustrate the definition.

- READE is a Manhattan partial street name for the valid Manhattan full street name READE STREET. READE STRE and READ are not partial street names, since they are formed by deleting portions of words rather than entire words.

- Both FRANCIS and FRANCIS LEWIS are Queens partial street names for the valid Queens full street name FRANCIS LEWIS BOULEVARD.

- PARK AVENUE is not considered a Manhattan partial street name, because it is a valid Manhattan full street name in its own right, even though it can be formed by deleting the last word from a valid Manhattan full street name, PARK AVENUE SOUTH.

- PARK is a Manhattan partial street name that can be formed from several valid Manhattan full street names, including PARK AVENUE, PARK AVENUE SOUTH, PARK ROW and PARK PLACE.

Geosupport accepts a partial street name as an input street name only if it unambiguously represents (i.e., if it can be formed only from) a single valid full street name in the specified input borough. If a partial street name can be formed from more than one full street name in the given borough, it is ambiguous and Geosupport rejects it. Consider the following examples:

- Several valid Manhattan full street names begin with the word PARK, as noted above. Therefore PARK is an ambiguous partial street name, and Geosupport does not accept it as an input street name for Manhattan. Similarly, two valid Manhattan full street names begin with the word YORK, namely YORK AVENUE and YORK STREET. Therefore, YORK is an ambiguous partial street name, and Geosupport does not accept it as an input street name for Manhattan.

- There is only one Manhattan street name that begins with the word READE, namely READE STREET. Therefore, Geosupport accepts the partial street name READE as a Manhattan input street name unambiguously representing the Manhattan full street name READE STREET.

- Geosupport accepts both FRANCIS and FRANCIS LEWIS as Queens input street names, since they are unambiguous partial street names for the Queens full street name FRANCIS LEWIS BOULEVARD.

- Some partial street names are accepted as input street names in some boroughs but not in others. For example, Geosupport accepts BROAD as an unambiguous partial street name for BROAD STREET in

Manhattan and in Staten Island. However, in Queens, BROAD is rejected as an ambiguous partial street name, since it can be formed from a number of different valid full Queens street names, including BROAD STREET and BROAD CHANNEL. In the Bronx and Brooklyn, BROAD is not a partial street name at all, and is rejected accordingly, since in those boroughs there are no full street names that begin with the word BROAD.

Note: Since street names may be added or deleted with each Geosupport release, the acceptability of partial street names may also change.

If a partial street name is accepted as an input street name, Geosupport returns the normalized version of the corresponding full street name in the WA1 output street name field, provided that the length of the normalized full street name does not exceed the SNL value that is in effect. If the length of the normalized full street name does exceed the SNL value in effect, Geosupport attempts to normalize the partial street name to fit within the SNL value; if that is successful, the normalized partial street name is returned in WA1. If neither the normalized full street name nor the normalized partial street name fits, Geosupport rejects the input as a street name that cannot be normalized within the SNL value in effect. If the SNL value in effect is 32 (the default value), it is certain that the normalized full street name will fit.

The following example illustrates the effect that varying the SNL value can have on street name normalizing. Suppose the input street name is CHAMBERS and the borough is specified as Manhattan. In this borough, CHAMBERS is accepted as an unambiguous partial street name for the full street name CHAMBERS STREET. If the SNL value in effect is 15 or greater, the output normalized street name is returned as CHAMBERS STREET . If the SNL is between 11 and 14 inclusive, the output street name is returned as CHAMBERS ST (the result of normalizing the full street name CHAMBERS STREET with an SNL of 11, 12, 13, or 14). If the SNL is between 8 and 10 inclusive, the partial street name CHAMBERS is returned. If the SNL is smaller than 8, the input is rejected as a street name that cannot be normalized within the current SNL value.

<u>III.5 The Similar Names Feature</u>

Geosupport has a ‘similar names’ feature that applications can utilize when handling Geosupport rejection of input street names. The feature consists of returning to the application a list of up to ten valid street names from the specified input borough that Geosupport deems to be ‘similar’ to the rejected input street name. Similar names are always full (not partial) street names, normalized in sort format. Applications can be designed to display the similar names whenever there are any, to aid the data entry operator in correcting rejected input names.

Whenever an input street name is rejected, if there is at least one valid full street name in the specified input borough that Geosupport deems to be similar to the rejected name, Geosupport takes the following actions:

- A list of the similar names, up to a maximum of ten, is returned in the List of Street Names field in WA1.

- The Geosupport Return Code value is ‘EE’. The Reason Code value is a number from 1 to 9 or the letter ‘A’, indicating the number of similar names that are in the List of Street Names. (The value ‘A’ indicates that there are 10 similar names.)

- An appropriate message is returned in the WA1 Message field.

- The number of similar names that are in the list is returned in the WA1 field Number of Street Names in List.

If there is exactly one similar name, the message explicitly indicates that name. For example, if the input is the invalid Manhattan name DUFFEY SQUARE, there is a single similar name, DUFFY SQUARE. The message in this case would be:

‘DUFFEY SQUARE’ NOT RECOGNIZED. IS IT ‘DUFFY SQUARE’?

If there is more than one similar name, then the message indicates the number of similar names but does not contain the similar names themselves. For example, the invalid Staten Island name ABBNER ROAD has three similar names. The message in this case would be:

‘ABBNER ROAD’ NOT RECOGNIZED. THERE ARE 003 SIMILAR NAMES.

To utilize the similar names feature, the user might program the application as follows.

- Whenever a call to Geosupport generates the GRC value ‘EE’, indicating rejection of an input street name and the existence of similar names, the application displays the Geosupport message (and/or the application’s own message) and the similar names. (When there is exactly one similar name, the Geosupport message already contains the similar name.)

- The application then offers the data entry operator an opportunity to correct the input name, either by selecting one of the similar names (for example, by allowing the operator to use the cursor and the ENTER key to make the selection) or by keying in a new name. If the operator has selected a similar name, the application moves it to the WA1 input street name field, overlaying the original input name, while leaving the rest of the WA1 input fields unmodified. The application then issues a second Geosupport call.

Designing the application to allow the operator to select a similar name from the list lessens the need for the operator to handle street name rejects by key-entering new street name spellings, thus increasing the operator’s productivity and eliminating the possibility of new key-stroke errors.

<u>Applications should never be designed to replace a rejected input name with a Geosupport-provided similar name in an automatic fashion, even when there is exactly one similar name</u>. The similar names that Geosupport provides are merely possibilities for the intended input street name, and it may well be that none of them is the intended input street name. Human judgment should always be exercised when deciding whether to use a similar name.

Although users need not be concerned with the criteria that Geosupport uses to generate similar names, the criteria are listed here for completeness. A valid full street name is deemed ‘similar’ to an invalid input street name if it is in the specified input borough and any of the following conditions holds:

(A) The valid full street name is at least as long as the input street name, and the two names are identical for the length of the input street name.

For example, in Manhattan, the valid full street names YORK AVENUE and YORK STREET would be deemed similar to the invalid name YORK. (YORK is invalid because it is an ambiguous partial street name.)

or

(B) There are no valid full street names in the specified borough that satisfy criterion (A), and the input street name begins with a compass direction word (NORTH, SOUTH, EAST or WEST) followed by a blank, and the input street name and the given valid full street name are identical up to and including the first three bytes following that blank.

For example, in Manhattan, consider the invalid input name EAST HOUSTIN STREET, which is 12 bytes long. For this name, there are no valid full Manhattan street names that satisfy criterion (A). That is, there are no valid full Manhattan street names that are longer than 12 bytes such that the first 12 bytes consist of the character string EAST HOUSTIN STREET. However, this input name begins with a compass direction word, EAST, and there is a valid full street name, EAST HOUSTON STREET, that is identical to EAST HOUSTIN STREET through the third byte following the blank after the word EAST (i.e., they are identical in their first eight bytes, consisting of the string 'EAST HOU'). Therefore, by criterion (B), EAST HOUSTON STREET is deemed similar to EAST HOUSTIN STREET.

or

(C) There are no valid full street names in the specified borough that satisfy criterion (A) or criterion (B), and the first three bytes of the input street name and the given valid full street name are identical.

For example, in Staten Island, each of the valid street names ABBEY ROAD, ABBOTT STREET and ABBY PLACE would be deemed similar to any invalid input name beginning with the characters ABB, such as ABBNER ROAD.

or

(D) The input street name contains numeric characters, and the input street name is identical to the valid street name up to and including the first numeric word.

For example, in Manhattan, the valid street name 8 AVENUE is deemed similar to the invalid name 8 PLACE. In Brooklyn, the valid street names BRIGHTON 6 COURT and BRIGHTON 6 STREET are both deemed similar to the invalid name BRIGHTON 6 AVENUE.

or

(E) In the boroughs of the Bronx and Manhattan only, all of the following are true:

(E1) The input street name can be transformed into the valid street name by adding the word EAST or WEST to the front of the input street name.

(E2) The input street name has at least two words.

(E3) The first word of the input street name is not END, RIVER, SIDE, ST or STREET.

(E4) The last word of the input street name is not EXTENSION.

The set of criteria in (E) is designed to reflect the common practice to specify street names of Bronx and Manhattan streets that begin with the word EAST or WEST without that first word. For example, the intersection of Broadway and West 42 Street in Manhattan is often expressed informally as "the intersection of Broadway and 42 Street"; pursuant to criteria (E), Geosupport generates EAST 42 STREET and WEST 42 STREET as similar names for the invalid Manhattan street name 42 STREET. Similarly, EAST HOUSTON STREET and WEST HOUSTON STREET are generated as similar names for the invalid Manhattan street name HOUSTON STREET. Criteria (E2) through (E4) filter out certain special cases where it is not customary to drop the first word EAST or WEST. For example, Criterion (E2) prevents the invalid Bronx input street name AVENUE from generating as similar names the valid Bronx street names EAST AVENUE and WEST AVENUE; Criterion (E3) prevents the invalid Manhattan street names END AVENUE, RIVER DRIVE and SIDE HIGHWAY from generating as similar names EAST END AVENUE and WEST END AVENUE, EAST RIVER DRIVE, and WEST SIDE HIGHWAY, respectively.

Note that, if the input street name is the invalid Manhattan name 7 STREET, then 7 AVENUE and 7 AVENUE SOUTH are similar names by virtue of criterion (D), and EAST 7 STREET is a similar name by virtue of (E), but WEST 7 STREET is not a similar name, since it is not itself a valid Manhattan street name.

The similar names are returned in the List of Street Names sorted in alphabetical order, except that any similar names that satisfy criteria (E) are listed first.

III.6 Unconventional Geographic Feature Names

In addition to conventional street names, Geosupport recognizes the following other types of geographic names: the names of 'paper streets'; the names of some non-street features; addressable and non-addressable place names; pseudo-street names; and intersection names. 'Recognizing' a name means that a street code has been assigned to that name and Geosupport accepts the name as an input name. The various types of unconventional names are discussed below, and there are further details on their processing in subsequent chapters.

Paper Streets

A paper street is a street that is legally 'mapped' (designated as a street on the official City Map) but that does not exist physically. The city 'maps' paper streets with the intention of constructing them, but there is no certainty that a particular paper street will be built. Indeed, some paper streets have been mapped and then eventually de-mapped without ever having been built.

Geosupport recognizes the names of paper streets, but it does not recognize geographic locations (addresses, intersections etc.) along a paper street. In addition to streets that are paper streets in their entirety, there are some streets that have both portions that exist physically and portions that exist only 'on paper'; for such a street, Geosupport recognizes geographic locations only within the portion that exists physically.

Non-Street Features

In the category of non-street features, as of this writing, Geosupport recognizes only the names of some railroad tracks and shorelines. Eventually, Geosupport will be enhanced to recognize the names of other non-street geographic features in New York City, including all railroad tracks and shorelines. Non-street features do not have addresses, but names of non-street features that are recognizable to Geosupport can serve as street name input to describe geographic locations other than addresses, such as intersections, street segments and street stretches.

Addressable Place Names

Addressable place names are the names of 'places', generally major individual buildings or building complexes, that can be combined with address numbers to form valid New York City addresses. Such places are not streets but their names serve the same role as do ordinary street names in forming addresses that Geosupport will recognize. An example in Manhattan that Geosupport recognizes is PENN PLAZA, a cluster of commercial buildings in the vicinity of Pennsylvania Station. For example, 1 PENN PLAZA, 2 PENN PLAZA and 7 PENN PLAZA are all valid Manhattan addresses, recognized by the U.S. Postal Service and by Geosupport's address processing functions. Other examples of addressable place names recognized by Geosupport are: in Manhattan, NEW YORK PLAZA, WASHINGTON SQUARE VILLAGE, GOVERNORS ISLAND and CONFUCIUS PLAZA; in Brooklyn, ALBEE SQUARE, METROTECH and FORT HAMILTON MANOR.

Non-Addressable Place Names (NAPs)

Non-Addressable Place names (NAPs) are names of buildings or other geographic features that cannot be combined with an address number to form a valid address. Note that a building that has a NAP may or may not also have a conventional street address; it is the place name that is non-addressable, not necessarily the place itself. For example, the Empire STATE BUILDING can be identified both by its name, which is a NAP, and by its conventional street address. CITY HALL in Manhattan and SHEA STADIUM in Queens are examples of NAPs referring to buildings that do not have conventional street addresses.

Typical geographic features that have NAPs include named buildings, stadiums, arenas, hospitals, housing projects, military complexes, museums, universities, theaters, airports, parks, zoos, marinas and islands. Geographic features that have NAPs are classified as either simplexes, complexes or constituent entities of a complex.

- A simplex is a monolithic named geographic feature, that is, a feature that has a NAP and is not a complex or a constituent entity of a complex. Examples in Manhattan: EMPIRE STATE BUILDING, CARNEGIE HALL, BRYANT PARK.

- A complex is a group of related geographically identifiable features at one site. A geographically identifiable feature is a feature that has an address, a NAP and/or a Building Identification Number (BIN). (BINs are discussed in detail in Section VI.3.) Examples of Manhattan complexes: LINCOLN CENTER, JEFFERSON HOUSES, CITY COLLEGE.

- A constituent entity of a complex is a building or other geographically identifiable feature that is part of a complex. A constituent entity may be identified by a NAP or by a conventional street address. Examples in Manhattan: AVERY FISHER HALL (a constituent entity of LINCOLN CENTER identified by NAP); CITY COLL SHEPARD HALL (a constituent entity of CITY COLLEGE identified by NAP); 259 CONVENT AVE (a conventional street address which identifies CITY COLL SHEPARD HALL).

NAPs are accepted as input data by Function 1N and by the address-processing functions (Functions 1, 1A and 1E). *Currently, these functions accept a limited set of NAPs (including only some of the examples in this section). Additional NAPs are being added over time.* For further details on NAPs, see Section IV.7.

Pseudo-Street Names

Pseudo-street names are special 'invented' names that in certain circumstances Geosupport accepts as valid input street names, as described in Sections V.2, VII.2 and VII.3. Three sets of pseudo-street names are:

DEAD END and its aliases DEADEND, DEAD END STREET, CUL DE SAC and CULDESAC

CITY LIMIT and its aliases CITY LIMITS and CITY LINE

BEND and its alias BENDING POINT

DEAD END and BEND, and their aliases, are valid in all five boroughs. CITY LIMIT and its aliases are valid only in the Bronx and Queens. These pseudo-street names may not be used to specify addresses, but they may be used to specify street intersections, and to specify the cross streets (but not the 'on' street) in other types of street configurations.

Another type of pseudo-street name that Geosupport recognizes, for certain addresses only, is Duplicate Address Pseudo-Street Names (DAPSs). DAPSs are used with Geosupport's duplicate address processing feature (discussed in detail in Section V.6). New York City has a small number of duplicate addresses, which are not data errors in Geosupport files, but are situations where an address is valid in reality at two different locations on the same street. DAPSs provide a means for a user to specify unambiguously a particular instance of a duplicate address.

An example of a street that has duplicate addresses is Hillside Avenue in Queens. A portion of Hillside Avenue in the Bellerose neighborhood has some of the same addresses as does another portion of Hillside Avenue in the Douglaston neighborhood. To make it possible to process these addresses, the DAPSs HILLSIDE AVENUE BELLEROSE and HILLSIDE AVENUE DOUGLASTON have been created. Similar DAPSs have been created for each city street that has duplicate addresses. In general, DAPSs are formed by augmenting the conventional name of the street with a neighborhood name.

Geosupport accepts DAPSs as valid input only for certain addresses on streets that have duplicate addresses, as explained in Section V.6. Geosupport never accepts DAPSs as input data for types of geographic locations other than addresses.

Intersection Names

Certain street intersections in New York City have intersection names, which serve as an alternative way to identify such an intersection in addition to the conventional means of reference using the names of two streets at the intersection. For example, ISAAC STERN PLACE is an intersection name for the intersection of West 57 Street and 7 Avenue in Manhattan. On the other hand, TIMES SQUARE is not an intersection name, because it refers to an area encompassing several intersections rather than a single street intersection. Official intersection names are designated by the City Council, and informal intersection names develop over time through local customary usage.

Geosupport accepts the input of a limited set of intersection names. Intersection names may not be used to specify addresses, but they may be used to specify street intersections, and to specify a cross street (but not the 'on' street) in other types of street configurations.

III.7 Street Name Browsing and Functions BB and BF

Functions BB ("browse backward") and BF ("browse forward") enable users to include interactive street name browsing functionality in their CICS applications. These functions may be used to assist data entry staff in determining valid spellings of street names that were rejected or the spelling of which is unknown to the staff. Functions BB and BF are supported in both the CICS and batch environments and are called using one work area.

A sequence of repeated calls to Functions BB and/or BF will browse backwards and/or forwards in alphabetical order through a list of all the valid normalized street names in a given borough. Each call to one of these functions returns up to ten names in alphabetical order. A call returns fewer than ten names if there are fewer than ten names remaining in the given borough in the given browse direction. When fewer than ten names are returned, a warning is issued. The starting point of the browse is determined by the value of the input character string.

Both Function BB and Function BF process an input borough code and character string, which are passed in the WA1 input Borough Code 1 and Street Name 1 fields, respectively. The input character string can be from one to 32 bytes long. The list of output normalized street names in alphabetical order is returned in the WA1 output field List of Street Names. The number of names returned is returned in packed decimal format in the MSW WA1 output field Number of Street Names in List. In COW WA1 output, the Number of Street Codes and Street Names in List is returned in character format.

The List of Street Names is a 320-byte WA1 output field containing ten 32-byte sub-fields or 'slots' for normalized street names. Let us call these sub-fields Namefield1 through Namefield10. (Do not confuse Namefield1 with the WA1 input street name field called Street Name 1.) Each output normalized street name is returned left-justified and blank-filled within its sub-field. When fewer than ten names are returned, the unused slots are left blank.

Function BF returns up to ten names for the given input borough, in alphabetical order, starting with the alphabetically first normalized name that is equal to or greater than the input character string. If the input string itself is a normalized name, it is returned in Namefield1, followed by the other returned names in Namefield2, Namefield3 etc., if any. Otherwise, the first name alphabetically greater than the input string for the given borough, if any, is returned in Namefield1, followed by the other returned names, if any.

Function BB works similarly but the list of up to ten names it returns ends with the alphabetically first name greater than or equal to the input string. Notice that, for a given input character string, there is an overlap of one name between the sets of names returned by Functions BB and BF.

If Function BB or BF returns ten names (in sub-fields Namefield1 through Namefield10) and the user wishes to continue the browse, additional browse function calls may be issued. For Function BF, prior to the subsequent call, the user primes the WA1 input field called Street Name 1 with the street name that was returned Namefield10. For Function BB, prior to the subsequent call, the user primes the WA1 input field called Street Name 1 with the street name that was returned in Namefield1.

CHAPTER IV: STREET CODES

IV.1 Introduction: Street Codes and Geographic Retrieval Consistency

This chapter discusses street codes, a set of numeric codes assigned in the Geosupport System to the city's street names and the names of certain non-street geographic features, place names, pseudo-street names and intersection names (see Section III.6). (In this chapter, except where otherwise stated, the terms 'street' and 'street name' refer to any geographic feature or feature name that has a Geosupport street code assigned to it.) Geosupport's street code feature provides critical support for many types of applications.

The primary purposes of the street code feature are:

- To enable applications to retrieve or match data from their own files by geographic location in a consistent manner: (See Section I.3 for a general discussion of the concept of geographic retrieval consistency.) Some streets have more than one name, and some street names have alternative spellings. Therefore, for applications that must retrieve data by types of geographic locations that are defined in terms of streets, such as addresses and intersections, the consistency of the retrieval is an important design consideration. For example, suppose a record is created in an application file for the Manhattan address 1204 SIXTH AVENUE. It is desirable that the application be able later to retrieve this record whether the user specifies the input address at retrieval time as 1204 SIXTH AVENUE, 1204 6 AVENUE or 1204 AVENUE OF THE AMERICAS. To achieve such consistency, Geosupport five-digit street codes rather than street names should be used in the retrieval key, as explained in this chapter.

- To obtain 'preferred' street names: For streets that have more than one name, the street name that is most appropriate to use for display purposes (such as on application screens, reports and mailing labels) may vary along the street. Street codes can be used to obtain location-specific 'preferred' street names for display, as explained in Section IV.6.

Secondary purposes of the street code feature are:

- To improve execution efficiency via street code input: There is an optional feature in which applications can pass input streets to Geosupport in the form of street codes rather than street names. This feature is useful when processing an application file that already contains street codes retained from a previous pass through Geosupport. The use of this feature can increase the execution efficiency of batch applications by sometimes allowing Geosupport to circumvent street name normalization and street code retrieval processing.

- To save application disk storage space: Storing street codes, instead of street names, in an application file saves application disk storage space. In many applications, however, doing so would necessitate increased programming and increased execution time overhead to make additional Geosupport calls to obtain street names for display.

The use of Geosupport street codes in an application does complicate the design and development of the application. It also adds a maintenance burden to the application, since street codes stored in an application file must be periodically resynchronized to reflect street code assignment changes effectuated in new Geosupport releases. In view of this overhead, the secondary purposes listed above are not likely by themselves to justify incorporating the use of street codes in an application.

IV.2 Street Name Relationships: Aliases and Locally Valid Street Names

GSS assigns street codes in a way that encodes certain information about street names. Specifically, a portion of the street code signifies whether an alias relationship exists between two street names; and a portion of the street code signifies whether a street name is only locally valid. These aspects of street code assignment can have implications for application design.

Two normalized street names are called aliases of each other if they are either alternative names of the same street (such as SIXTH AVENUE and AVENUE OF THE AMERICAS in Manhattan) or any portion thereof, or are spelling variants of the same street name (such as SIXTH AVENUE and 6 AVENUE, or MAC DOUGAL STREET, MACDOUGAL STREET and MCDOUGAL STREET). Geosupport is designed to recognize all commonly accepted street name aliases, and through the structure of its street code assignments, to identify whether two street names are aliases for the same street.

Locally valid street names are street names that are only valid 'locally', that is, for a portion of a street. Almost all streets that have locally valid street names also have at least one name that is valid for the entire street. An example is Seventh Avenue in Manhattan, which has the following names:

- The names 7 AVENUE and SEVENTH AVENUE are valid for the entire street.
- POWELL BOULEVARD and various aliases (ADAM CLAYTON POWELL JR BOULEVARD, A C POWELL BOULEVARD etc.) are valid only for the portion of the street north of Central Park.
- FASHION AVENUE is valid only for a portion of the street in the Garment District.
- SAINT VINCENTS SQUARE and ST VINCENTS SQUARE are valid only for a small stretch of the street in the vicinity of Saint Vincent's Hospital.

All of the above names are aliases of each other, since they are all names for the same street or a portion thereof. The names in the first set are valid for the entire length of the street; the other names are only valid locally. Notice that two street names can be considered aliases of each other even if there are no locations at which both names are valid. For example, FASHION AVENUE and SAINT VINCENTS SQUARE are aliases, even though there is no location where both names are valid.

IV.3 Five-Digit and Ten-Digit Street Codes

To each normalized spelling of a full street name within a borough, a ten-digit number called the ten-digit street code (10SC) is assigned. Partial street names (see Section III.4) are assigned the same 10SC values as the full streets names from which they were generated.

A 10SC value is meaningful only within a borough, and is generally preceded by a borough code to form an eleven-digit item called the borough and ten-digit street code (B10SC). If two street names in different boroughs happen to have the same 10SC value, that does not signify any relationship between those streets. **Streets in two different boroughs are always considered to be different streets,** even if the two streets have the same name, and even if they form a single physically continuous street running across the borough boundary. For example, Atlantic Avenue crosses the Brooklyn-Queens border. Geosupport treats the Brooklyn and Queens portions of Atlantic Avenue as two different streets, each with its own B10SC value ('31343001010' and '42889001010', respectively).

The first five digits of the 10SC are called the five-digit street code (5SC). The 5SC has a fundamental significance: **the 5SC values of two street names in a borough are identical if and only if those names are aliases for the same street**. Positions six through ten of the 10SC are discussed in Section IV.5.

Like the 10SC, the 5SC is meaningful only when accompanied by a borough code; when concatenated, the borough code and 5SC form a six-byte item called the borough and five-digit street code (B5SC). The B5SC simply consists of the first six bytes of the B10SC. For MSWs only, Geosupport sometimes represents the B5SC as a four-byte packed decimal item, referred to as the packed borough and five-digit street code (PB5SC).

Conceptually, a B10SC value represents a particular (normalized) spelling of a particular name for a street within a borough, while a B5SC value represents the street itself and is shared by all the street's aliases. Consider the following examples of Manhattan street names, grouped by street, i.e. by five-digit street code. (Note: '1' is the borough code for Manhattan.)

(Normalized) Street Name	B10SC =	B +	5SC +	Remainder of 10SC
5 AVENUE	11041001010	1	10410	01010
FIFTH AVENUE	11041001020	1	10410	01020
MUSEUM MILE	11041002010	1	10410	02010
6 AVENUE	11051001010	1	10510	01010
SIXTH AVENUE	11051001040	1	10510	01040
AVENUE OF THE AMERICAS	11051001030	1	10510	01030
7 AVENUE	11061004010	1	10610	04010
SEVENTH AVENUE	11061004020	1	10610	04020
FASHION AVENUE	11061002010	1	10610	02010
POWELL BOULEVARD	11061001080	1	10610	01080
A C POWELL BOULEVARD	11061001010	1	10610	01010
7 AVENUE SOUTH	11071001010	1	10710	01010
SEVENTH AVENUE SOUTH	11071001020	1	10710	01020
EAST 21 STREET	11741001010	1	17410	01010
EAST 21	11741001010	1	17410	01010
WEST 21 STREET	13419001010	1	34190	01010
WEST 21	13419001010	1	34190	01010

The above example illustrates several aspects of street code assignment. Notice that alias names of the same street have the same B5SC value. Notice that EAST 21 STREET and WEST 21 STREET have different B5SC values, which amounts to treating them as names of two different streets (as indeed they must be treated, since they have address numbers in common). Similarly, 7 AVENUE and 7 AVENUE SOUTH are treated as two different streets. Notice that partial street names have the same B10SC's as the full names from which they were generated, such as EAST 21 and EAST 21 STREET.

By using B5SC's in the retrieval key instead of street names, applications can achieve consistent retrieval or matching of application data by types of locations involving streets. We outline below how an application might be designed for consistent retrieval or matching by address, which requires a retrieval key consisting of a B5SC and a normalized address number (discussed in Section V.2). When the type of location being retrieved involves more than one street, such as intersections, the key would be designed to contain a B5SC field for each street.

- At record creation time: During the initial creation of a record in the application file, the application calls Geosupport to obtain the B5SC corresponding to the input street name, as well as the normalized form of the input address number. The application uses these items to form a geographic retrieval key, which it stores in the new application record. Two files that contain such a key can be matched directly on the key, resulting in a match that will be consistent, i.e. independent of the use of street name aliases.

- At retrieval time: When retrieving data from the application file by address, the application again calls Geosupport, obtaining the B5SC and normalized address number corresponding to the input street name and address number. The application formats these items into a search key, and reads the application file using this key. The use of the B5SC in the key instead of the street name allows the retrieval to be consistent, i.e. independent of which alias for the street is passed as input.

Geosupport has three display functions, Functions D, DG and DN, which can be used to obtain street names for display in application screens, reports, mailing labels etc. These functions process five-, seven- and ten-digit street code input, respectively. Section IV.6 discusses the display functions.

IV.4 Resynchronization of Street Codes Stored in User Files

When designing an application in which street codes, either B5SCs or Bl0SCs, are stored in an application file, the user must consider the important issue of resynchronizing those street codes with respect to new Geosupport releases. This issue arises because it is sometimes necessary for the GSS staff to change the B5SC value (and therefore also the Bl0SC value) that is assigned to a street name. This would be necessary if they determine that two street names that currently have different B5SC values (signifying that they are names of two different streets) are in reality aliases for the same street and therefore must be made to have the same B5SC value. Conversely, the GSS staff might determine that two names that currently have the same B5SC value are actually names of two different streets, and therefore must be made to have different B5SC values. Both of these types of problems can be rectified only if the B5SC value, and therefore the B10SC value, assigned to one or more street names is changed.

Whenever a new Geosupport release is implemented that includes any changes to street codes assigned to street names, it is essential for users to make the corresponding changes to all occurrences of those street codes stored in application files. This street code ‘resynchronization’ should be timed to be as simultaneous as possible with the implementation of the new Geosupport release. User failure to resynchronize the street codes stored in an application file for each new Geosupport release could have serious negative consequences for the application. Geographic searches in the application file in which the street code is used as part of the retrieval key might fail to retrieve some application records or might retrieve inappropriate ones. Matching of records within an application file or between files by geographic location could fail, or could result in an erroneous match. Also, the display functions D, DG and DN could return inappropriate street names for some street code values.

Prior to the implementation of each new Geosupport release, a Street Name/Street Code Change Bulletin is sent to all users listing the street code changes being made in the new release. Also, as part of each release, a Geosupport file called the Street Code Change File (SCCF) is created and made accessible to users. If an application file contains B10SCs, the user can utilize the SCCF to develop an automated batch resynchronization procedure. Of course, when the stored B10SCs are updated, that also updates the B5SCs that comprise the first six bytes of the B10SCs. (If for some reason there is a separate B5SC field in the application file in addition to a B10SC field, the B5SC field must be overlayed with the new value using the first six bytes of the new B10SC value.) Resynchronizing stored B10SCs using the SCCF is the optimal resynchronization method. <u>Users are strongly urged, when designing new aplications in which street codes are to be stored in application files, to design those files so they contain B10SCs, either in addition to or instead of street names, and to write a batch street code resynchronization program that uses the SCCF.</u>

In existing applications in which B10SCs are not currently stored in the application file, we recommend that the file be enhanced to contain B10SCs, enabling a resynchronization procedure that uses the SCCF to be developed. If the file currently contains street names, B10SCs can be inserted into it easily using Function 1N (discussed in Section III.2). If only B5SCs are currently stored in the file, not street names, a more difficult one-time effort would be required to insert B10SCs into the file; the methodology for doing this would be similar to the resynchronization procedure using B5SCs described below. For application files that do not currently contain B10SCs and cannot be enhanced to contain them, the SCCF cannot be used to resynchronize the B5SCs, and other resynchronization methods must be used.

The various methods for resynchronization are discussed below. <u>It is the user's responsibility to develop a street code resynchronization procedure for each application file in which street codes are stored, and to run that procedure as soon as possible after each new Geosupport release is placed into production.</u>

<u>Resynchronization procedure using B5SCs</u>

When only B5SCs are stored in the application file, not the original input street names nor B10SCs, it is not possible to develop a fully automated procedure to resynchronize those B5SCs. Instead, records in the application file that are affected by street code changes (as listed in the Street Name/Street Code Change Bulletin) must be found and individually examined and updated. This is because of the inherent ambiguity of a B5SC value, which can be associated with more than one street name. Specifically, it is possible that two or more street names that had the same B5SC value prior to the new Geosupport release no longer have the same value in the new release. When this occurs, the user cannot resynchronize the old B5SC value mechanically, but must determine, for each occurrence of the old B5SC value in the application file, which street name that occurrence represents, in order to determine what the new B5SC value should be for that occurrence. In order to make that determination, the user would have to research individually each record containing such a B5SC value, using any information that could help to pinpoint the location and thus to determine whether the B5SC value should be changed and what the new value should be. Such information as an address, cross streets, a zip code, a community district or other district identifier, or tax block and tax lot identifiers could be helpful for this purpose. Because this procedure is not automatic, it is the least desirable method.

<u>Resynchronization procedure using street names</u>

If the application file contains the original input street names in addition to B5SCs, then the user can develop a fully automated batch procedure for resynchronizing the B5SCs, albeit a less than optimal one, as follows. The user can write a batch program that calls Function 1N to obtain, for each original input street name, the B5SC value currently (in the new Geosupport release) assigned to that name. The program would process every record in the application file, automatically replacing the B5SC value already stored in each record with the current B5SC value obtained from Function 1N. The program would have to provide for handling any Function 1N rejects, that is, street names that are no longer valid in the new Geosupport release.

Resynchronization procedure using B10SCs

Using stored street names to resynchronize B5SCs is preferable to using just the B5SCs themselves, because the former method can be automated while the latter cannot. Nevertheless, the former method is highly inefficient, because it necessitates processing every record in the application file, even though in each Geosupport release only a tiny portion (if any) of the city's street names have street code assignment changes.

Storing B10SCs in the application file, and using the SCCF to resynchronize them, is the optimal approach to street code resynchronization. The ambiguity intrinsic to B5SCs does not exist for B10SCs. Since every B10SC value is assigned to a single street name only, stored B10SCs can be resynchronized by automatically replacing every occurrence of each changed B10SC value with the proper new value, with no research required to determine the latter.

GSS creates a new release of the SCCF as part of each new Geosupport release. The following DD statement gives users batch access to the SCCF:

```
//ANYDDNM  DD  DSN=A030.STREET.SCCF,DISP=SHR
```

The SCCF is a sequential file with 80-byte records. The SCCF has a single header record containing file identification information, followed by a set of data records. The layouts of the header and data records are as follows:

Street Code Change File (SCCF) Record Layouts

Header Record

Field	Size	Positions	Comments
Header constant	42	1 - 42	Literal constant: 'GEOSUPPORT SYSTEM FOREGROUND HEADER RECORD'
DDNAME of File	8	43 - 50	Literal constant: 'SCCF '
Geosupport Release Identifier	4	51 - 54	e.g. '04A ' (4th byte is generally blank)
Date of File Creation	6	55 - 60	yymmdd format
Filler	20	61 - 80	

Data Records

Field	Size	Positions
Old B10SC	11	1 - 11
Filler	5	12 - 16
New B10SC	11	17 - 27
Filler	5	28 - 32
Borough Code	1	33
Filler	1	34
Street Name	32	35 - 66
Filler	14	67 - 80

(The fields Borough Code and Street Name are in the SCCF records for informational purposes only and are not needed for synchronization.)

In an application file containing stored B10SCs, the first six bytes of the B10SC field (or fields, if the geographic location represented in the record involves more than one street, such as records for intersections) constitute the B5SC field, which could be defined as part of a key for consistent geographic retrieval; therefore, a separate B5SC field would not be needed for that purpose. The full B10SC field could also itself be defined as a direct access key, for use by the resynchronization program.

The resynchronization program would read the SCCF sequentially. For each SCCF record, the program would read the application file directly using the old B10SC value in the SCCF record as the search key. All occurrences of this B10SC value found in the application file would be replaced by the new B10SC value from the SCCF record. (Note: most application files contain multiple records for the same street. Hence, if the application file is a VSAM file, in most applications, the B10SC field(s) must be defined as an alternate key(s), not as the primary key, since several records could have the same key values. In addition, since the resynchronization program modifies a key value, the UPGRADE option should be specified in the DEFINE ALTERNATE INDEX component of the IDCAMS control file. Similar considerations might apply for other types of direct access files.)

Summary of Street Code Resynchronization

The resynchronization of street codes stored in an application file is an important issue for application design. If the application must retrieve records by geographic location, it is necessary to use B5SCs in the retrieval key in order to make the retrieval geographically consistent, and therefore B5SCs must be stored in the application file. Since the street codes that are assigned to some street names can be changed in new Geosupport releases, these stored B5SCs (and/or stored B10SCs, if any) must be synchronized to reflect these changes. However, as we have seen, the synchronization of B5SCs cannot be fully automated unless either the originally entered street names or the B10SCs corresponding to those names are stored in the record.

Storing street names requires considerable disk storage space, and resynchronization using street names is inefficient, since every street name in the application file would have to be processed. The best alternative from the points of view of both disk storage efficiency and resynchronization efficiency is to store B10SCs in the application file. In that case, the first six bytes of the B10SC field, which is the B5SC, could be defined as part of a geographic retrieval key. For resynchronization, the full B10SC could be used in conjunction with the Street Code Change File (SCCF). The user could develop a highly efficient procedure in which only those application records containing B10SC values that are in SCCF records, i.e., that have been changed in the new Geosupport release, would be accessed and updated.

In view of the above considerations, the following design guidelines are strongly recommended with respect to resynchronizing street codes stored in application files:

- When an application is being designed in which there will be a file to which consistent street-related geographic access is required, then the file should be designed to contain B10SCs obtained from Geosupport.

- Application programs that access the file geographically should be written to use the first six bytes of the B10SC field, which constitute the B5SC, in the geographic retrieval key. In existing application files that contain B5SCs but not B10SCs, the B5SC field should be enhanced into a B10SC field. The B5SC portion of the B10SC should continue to be used for geographic retrieval. The entire B10SC field should be defined as a direct access key, to support efficient automated street code resynchronization.

- The user should develop a batch procedure to resynchronize the B10SCs stored in the file, using the SCCF. An optimal procedure would access directly those records in the application file that contain B10SCs for which there are changes in the new Geosupport release (i.e., for which SCCF records exist). The user should run this procedure routinely whenever a new Geosupport release is implemented.

IV.5 Seven-Digit Street Codes: Local Street Name Validity, Local Group Codes

This section discusses local street name validity, the phenomenon that some street names are valid for only a portion of the street. (In fact, there are even a few New York City streets that do not have a single street name that is valid for the street's full length.) A street that illustrates the phenomenon of local street name validity is Seventh Avenue in Manhattan. The names 7 AVENUE and SEVENTH AVENUE are valid for the full length of the street. The name ADAM C POWELL BOULEVARD and its various spelling variants (POWELL BOULEVARD, A C POWELL BOULEVARD etc.) are valid only for the portion of the street north of Central Park. The name FASHION AVENUE is valid in the Garment District. The names SAINT VINCENTS SQUARE and ST VINCENTS SQUARE are valid for a portion of the street in the vicinity of Saint Vincent's Hospital.

All of the Geosupport functions that accept street name input, except for Function 1N, perform local street name validation, which verifies whether the input street name is specifically valid for the input location, and if it is not valid, returns up to four aliases of the invalid name that are valid for the given location. Local street name validation is performed automatically when a two-work-area call is made, with no special user action required to invoke it.

If, in a two-work-area call, the input street name is not valid for the given input location, Geosupport takes the following actions:

- The call is rejected with a Geosupport Return Code of '50' and an appropriate Message.

- Up to four locally valid street name aliases of the invalid street name are returned in the WA1 List of Street Names field. (Specifically, the names that are returned are the principal street name of each local street name group that is valid for the given location. The concepts of 'principal street name' and 'local street name group' are discussed below and in the next section.)

- The Reason Code contains the number of names returned in the List of Street Names.

- For MSW: the WA1 Number of Street Names Field contains the number of names returned in the List of Street Names, in packed decimal format. For COW: the WA1 Number of Street Codes and Names Field contains the number of names returned in character decimal format.

For example, the address 375 7 AVENUE in Manhattan is located south of Central Park, so the street name ADAM C POWELL BOULEVARD is invalid for this address. Therefore, a two-work-area call to Function 1E (for example) with the input address as 375 ADAM C POWELL BOULEVARD would result in a GRC '50' rejection and the return of the locally valid alias street name 7 AVENUE in the List of Street Names. On the other hand, 2019 7 AVENUE is north of Central Park, so 2019 ADAM C POWELL BOULEVARD is a valid address, and is accepted by Function 1E. Similarly, a two-work-area call to Function 2 would reject the intersection of ADAM C POWELL BOULEVARD and WEST 56 STREET with a GRC of '50' and the return of the locally valid alias street name 7 AVENUE in the List of Street Names.

Local Group Codes (LGCs)

Local street name validity is reflected in the sixth and seventh digits of the 10SC, which constitute the Local Group Code (LGC). Street codes are assigned in such a way that two names for a street have the same LGC value if and only if those names are valid for the same portion (possibly all) of the street. Note that if two names are valid for overlapping portions of a street, or one is valid for a subset of the portion where the other is valid, then those names are in different local groups. In order to be in the same local group, names must be valid for exactly the same portion of the street.

Conceptually, the set of all street names for a given street can be viewed as being partitioned into subsets called 'local street name groups', each group identified by its LGC value and consisting of all the names that are valid for a particular portion (possibly all) of the street. (Most New York City streets only have one local street name group.)

A LGC value is meaningful only relative to its B5SC value. The B5SC identifies the street, and the LGC identifies a local street name group for the given street, that is, the group of all names for the given street that are valid for a particular portion (possibly all) of the street. The B5SC concatenated with the LGC, that is, the first eight bytes of the B10SC, constitute the Borough and Seven-Digit Street Code (B7SC). Two street names have the same B7SC value if and only if they are names for the same street (same B5SC value) and are valid for the same portion of the street (same LGC value relative to the given B5SC value).

Street Name Codes (SNCs)

The final three digits of the B10SC are called the Street Name Code (SNC). Thus, the B10SC consists of the concatenation of the borough code, 5SC, LGC and SNC. The SNC serves simply to serialize the street names within a local group, so that the full B10SC is unique to a specific spelling of a specific street name.

Consider the example of Seventh Avenue in Manhattan. The following is a list of many of Geosupport's normalized aliases for this street, sorted by B10SC. The dashed lines highlight the four local street name groups.

STREET NAME	B10SC =	BORO+	5SC +	LGC +	SNC
A C POWELL BOULEVARD	11061001010	1	10610	01	010
AC POWELL BOULEVARD	11061001020	1	10610	01	020
ADAM C POWELL BOULEVARD	11061001030	1	10610	01	030
ADAM POWELL BOULEVARD	11061001040	1	10610	01	040
ADAM POWELL JR BOULEVARD	11061001050	1	10610	01	050
ACP BOULEVARD	11061001060	1	10610	01	060
A C P BOULEVARD	11061001070	1	10610	01	070
POWELL BOULEVARD	11061001080	1	10610	01	080
ADAM CLAYTON POWELL BOULEVARD	11061001090	1	10610	01	090
ADAM CLAYTON POWELL JR BOULEVARD	11061001100	1	10610	01	100
FASHION AVENUE	11061002010	1	10610	02	010
SAINT VINCENTS SQUARE	11061003010	1	10610	03	010
ST VINCENTS SQUARE	11061003020	1	10610	03	020
7 AVENUE	11061004010	1	10610	04	010
SEVENTH AVENUE	11061004020	1	10610	04	020

All of the names in the above list are aliases of each other, and therefore their B10SCs have the same B5SC value, '110610'. The first ten names in the list, A C POWELL BOULEVARD and its nine spelling variants, constitute the group of names valid only for the part of the street north of Central Park; this local group is identified by LGC value '01' and B7SC value '11061001'. The name FASHION AVENUE is valid only for the portion of the street in the Garment District, and constitutes the sole member of local group '02'. Local group '03' consists of the names SAINT VINCENTS SQUARE and ST VINCENTS SQUARE, which are valid only for a small portion of the street in the vicinity of Saint Vincent's hospital. Finally, the names 7 AVENUE and SEVENTH AVENUE are valid for the entire street and constitute local group '04'.

IV.6 Functions D, DG and DN; Primary, Principal and Preferred Street Names

This section discusses Functions D, DG and DN. These functions are referred to as the Geosupport 'display functions' because, although they do not actually display anything themselves, they return street names that applications can use to format geographic locations for display on reports, screens, mailing labels, work orders for field work etc. Functions D, DG and DN process input five-, seven- and ten-digit street codes (accompanied by borough codes), respectively. (The display functions can also be used to obtain address numbers in display format. This is discussed in Section V.2)

The selection of street names for display is a significant consideration for any street that has more than one local street name group. Given a specific location (i.e., an address, intersection, street segment or block face) on a street, applications can use a simple procedure involving a call to Function DG to obtain a street name that is considered 'optimal' to display for that location, called the 'preferred street name'. Functions D and DN return street names that are of more specialized and limited use.

The display functions do not have a Work Area 2, and are accessed via one-work-area calls only. (If a second work area is mistakenly supplied, it is ignored.) Work Area 1 contains fields for the input street codes and for the output street names. The street names that the display functions return are full street names (never partial street names), normalized in accordance with the SNL and Street Name Normalization Format Flag values that are in effect for the given call.

Input Data

For the convenience of applications, each of the display functions can process up to three input street codes in a single call. If there are multiple input street codes, they are processed independently of each other and are not treated as though they were specifying a geographic location such as an intersection. (Similarly, if there is both an input street code and an input address number, they are not treated as though they were specifying an address.) In particular, a successful call to a display function does not imply the validation of the input data as a geographic location. Furthermore, the output street names returned by the display functions are not by themselves customized to be location-specific. Function DG must be used in conjunction with a call to another function to obtain a location-customized preferred street name.

For Function D, there are two different sets of fields in Work Area 1 that applications can use to pass input street code values, as follows.

- For MSW calls, if the input street codes are in the form of PB5SCs, the 4-byte input fields that are labeled PB5SC-1, PB5SC-2 and PB5SC-3 in the Work Area 1 layout in Appendix 2 are used.

- For MSW or COW calls, if the input street codes are in the form of B5SCs, the 11-byte input fields labeled B10SC-1, B10SC-2 and B10SC-3 are used. The input B5SC values, which are 6 bytes long, must be passed in these fields left-justified, and the contents of these fields beyond the first six bytes are ignored.

If an MSW application passes values to Function D in both sets of input street code fields (presumably inadvertently), the PB5SC fields are processed, and the B10SC fields are ignored.

For Function DG and DN, only the fields B10SC-1, B10SC-2 and B10SC-3 are used. In the case of Function DG, input 8-byte B7SC values must be passed in these fields left-justified, and the contents of these fields beyond the first eight bytes are ignored.

In a call to any of the display functions, the input street code fields must be used without 'skipping'. That is, if an input street code field contains blanks, subsequent input street code fields of the same type are ignored. Thus, if PB5SC-1 is blank, then PB5SC-2 and PB5SC-3 are ignored; if PB5SC-1 is not blank but PB5SC-2 is blank, then PB5SC-3 is ignored. Similar conditions hold for the B10SC fields.

Output Data

The output street names that the display functions return are as follows:

- Function D is used to obtain, for an input B5SC value, or PB5SC value (MSW only), the primary street name for the given street. The primary street name is one alias, that is, one spelling of one street name, that GSS has designated, from among all the aliases for the street, as 'best' representing the street as a whole. (Note: The designation of primary street names has no 'official' status, and of necessity sometimes involves an element of arbitrariness.) The primary street name is not customized to be the 'best' name for any particular location along the street; it is simply the street name deemed most suitable to display if a single street name must be used to represent the entire street. Most applications do not require the services of Function D. However, some applications may, for example, have a requirement to display a consistent street name for all locations that are on the same street, so that it will be clear to users that all the displayed locations do refer to the same street. When possible, GSS designates as primary a street name that is valid for the entire length of the street. However, it is important to note that there are a few streets that do not have any such names. **On such streets, there are locations where the street's primary street name is invalid.**

- <u>Function DG</u> is used to obtain, for an input B7SC value, the <u>principal street name</u> of the corresponding local street name group. This is a street name belonging to the given local street name group that GSS has designated as 'best' representing that group of street names, that is, the name that has been deemed to have the most 'standard' spelling. (As with primary street names, the designation of principal street names has no 'official' status, and of necessity sometimes involves an element of arbitrariness.) The most important use of Function DG is to retrieve preferred street names, as discussed below.

- <u>Function DN</u> is used to obtain, for an input B10SC value, the full street name spelling to which that B10SC value uniquely corresponds. Function DN is useful mainly in certain atypical applications that store ten-digit street codes in an application file, but do not store the input street names from which the street codes were originally obtained, and the application has a requirement (for legal purposes, for example) to display those originally-entered street names. Such applications can use Function DN to obtain the original name from the corresponding stored B10SC value (although the name will be provided in normalized form.) Application designers can obviate the need to make Function DN calls by retaining in the application file either the original input street name or that name in normalized form.

The display functions return one output street name for each valid input street code. For each input street code that is invalid, the display functions return all question marks (the character '?') in the corresponding output street name field. In addition, if at least one input street code is invalid, the GRC value '64' is issued along with an appropriate Message.

<u>Preferred Street Names</u>

As explained above, the primary street name is not necessarily the 'best' name to use to express any particular location along a street. Furthermore, although each principal street name is the 'best' representative of its local group of street names, there may be more than one local group valid at a particular location.

Given a specific address, street intersection, street segment or block face along a street, applications can use Function 1, 2, 3 or 3C, respectively, in conjunction with Function DG, to obtain the preferred street name specific to that location, as follows:

1. Issue a two-work-area call to the appropriate location-processing function (Function 1, 2, 3 or 3C), to obtain an item called the 'DCP-preferred LGC', which all of these functions return in WA2. The DCP-preferred LGC represents the block face-specific local street name group that GSS has designated from among those local groups that are valid for the given block face as being the 'best' group of street names to display for that block face. (Note: In support of unique requirements of the New York City Board of Elections' voter registration application, Function 1E returns an item called the BOE-preferred LGC in place of the DCP-preferred LGC. For most addresses, the DCP-preferred LGC and the BOE-preferred LGC are identical.)

2. Concatenate the DCP-preferred LGC to the B5SC to form a B7SC.

3. Call Function DG with the above B7SC as input to obtain the preferred street name. The principal street name of the DCP-preferred local group is the preferred street name for the given location.

For example, suppose the original user input address to an application is 2019 SEVENTH AVENUE in Manhattan. This address is within the portion of the street north of Central Park, where two local groups are valid: local group 1 (LGC = ‘01’), which consists of the name POWELL BOULEVARD and its spelling variants, and local group 4 (LGC = ‘04’), which consists of the names 7 AVENUE and SEVENTH AVENUE. GSS has designated local group 1 as the DCP-preferred LGC for the portion of the street north of Central Park, and has also designated ADAM C POWELL BOULEVARD as the principal name of this local group. To obtain the preferred street name for the address 2019 SEVENTH AVENUE in Manhattan, the procedure outlined above would be performed, as follows:

- Function 1 is called with the input address 2019 SEVENTH AVENUE. Function 1 returns in WA1 the B10SC value of the input street name, SEVENTH AVENUE, namely, the value ‘11061004020’. The first six positions, ‘110610’, constitute the street’s B5SC value. Function 1 also returns, in WA2, the DCP-preferred LGC value for this address, which is ‘01’.

- The application concatenates the B5SC value with the DCP-preferred LGC value, forming the B7SC value ‘11061001’.

- The application calls Function DG using this B7SC value as input, obtaining ADAM C POWELL BOULEVARD as the preferred street name corresponding to the address 2019 SEVENTH AVENUE. The application may now display the address as 2019 ADAM C POWELL BOULEVARD.

In summary, the application began with the address 2019 SEVENTH AVENUE, and by following the outlined procedure, in which first Function 1 and then Function DG was called, the application formed the ‘preferred’ address 2109 ADAM C POWELL BOULEVARD for display. If the application had started with the address 375 SEVENTH AVENUE, which is located south of Central Park, the procedure would have resulted in the formation of the address 375 7 AVENUE, since 7 AVENUE is the principal name of the DCP-preferred local group for all locations on Seventh Avenue south of Central Park.

IV.7 Street Codes and Non-Addressable Place Names

This section discusses the manner in which street codes are assigned to Non-Addressable Place names (NAPs). Every NAP is a name of a simplex, a complex or a constituent entity of a complex (see Section III.6). By definition, every simplex and every complex has a NAP. A constituent entity of a complex may or may not have a NAP, and may or may not have one or more addresses. (A building that has neither an address nor a NAP is called a Non-Addressable Un-named Building (NAUB). NAUBs can be identified only by their Building Identification Numbers (BINs). NAUBs and BINs are discussed in detail in Chapter VI.)

Like conventional street names, some NAPs have aliases (alternative names and spelling variants). For example, CABRINI MEDICAL CENTER and CABRINI MED CENTER are spelling variants of the same name; AVERY FISHER HALL and PHILHARMONIC HALL are alternative names of the same geographic feature. As with conventional street names, the B10SC values assigned to NAP aliases have the same B5SC value.

In the case of a complex and its constituent entities, the assignment of street codes is analogous to the methodology used for streets. In general (the exceptional case is described below), the names of the entire complex and the names of its constituent entities are all treated as aliases of each other (that is, their B10SCs have the same B5SC value), since they are all names of the same geographic feature (the entire complex) or parts thereof (the constituent entities of the complex). Within the umbrella of this B5SC value, the NAPs that are valid for each portion of the complex, namely, either the entire complex or a particular constituent entity, are assigned to a different local group. Thus, the entire complex has its own distinct B7SC value, and each constituent entity has its own distinct B7SC value.

The following NAPs associated with Manhattan's Lincoln Center complex illustrate the assignment of street codes to NAPs associated with a complex.

NAP	B 10SC
LINCOLN CENTER	1 25006 01 010
LINCOLN CTR FOR THE PERFRMG ARTS	1 25006 01 030
NY STATE THEATER	1 25006 02 020
NEW YORK STATE THEATER	1 25006 02 040
N Y STATE THEATER	1 25006 02 060
AVERY FISHER HALL	1 25006 03 010
PHILHARMONIC HALL	1 25006 03 030

All of the NAPs associated with Lincoln Center (of which only a sample is listed above) have the same B5SC value, 125006. Within this B5SC value, the LGC value 01 is assigned to the NAPs of the complex as a whole, LINCOLN CENTER and LINCOLN CTR FOR THE PERFRMG ARTS (and other variants not listed). The LGC value 02 is assigned to NY STATE THEATER and variants thereof, and so on. An application can use the B5SC value, 125006, as the retrieval key to retrieve all the records in an application file for the NAPs associated with Lincoln Center, both records for the complex as a whole and records for its constituent entities. If only the records for the complex as a whole are to be retrieved, the application would use the B7SC value 12500601. If only the records for the New York State Theater are to be retrieved, the application would use the B7SC value 12500602, and so on. (Note that none of these retrievals would retrieve any records stored by address, since the street name or addressable place name in an address would have a different B5SC value than the one assigned to the NAPs.)

The exception to the method of street code assignment described above occurs in the case of a complex that has 99 or more constituent entities that have their own NAPs. *[As of this writing, no instances of this case have been implemented.]* Since the LGC is a two-byte numeric item between '01' and '99', a B5SC value is limited to having at most 99 local groups. In the exceptional case, the constituent entities that have NAPs are too numerous to accommodate within a single B5SC value. Therefore, these NAPs are gathered into sets of up to 99 NAPs each. A different B5SC value is assigned to each set. Within each set, each NAP, along with its true aliases, is assigned its own LGC.

IV.8 Street Code Input Feature

For most functions involving street input, applications have the option to pass the input streets to Geosupport in the form of either street names or street codes. (The exceptions are Function 1N and the display functions. Function 1N requires input streets to be passed in the form of street names, since that function's sole purpose is to normalize input street names and provide their street codes. The display functions require input streets to be passed in the form of street codes, since those functions are designed to provide street names corresponding to input street codes.)

The street code input feature is useful in an application that stores street codes but not street names in an application file (presumably to save disk storage space), since it enables the application to process records from that file directly through a Geosupport location-processing function, without first having to call a display function to obtain street names.

Applications can provide input street codes to any of the functions that can accept them in any of the following forms (the field names used below are the same as those used in the WA1 layouts in Appendix 2 for MSWs and Appendix 13 for COWs.):

- PB5SCs (MSW only), passed in as many as necessary of the WA1 input fields PB5SC-1, PB5SC-2 and PB5SC-3.

- B5SCs, passed left-justified and space-filled in as many as necessary of the WA1 input fields B10SC-1, B10SC-2 and B10SC-3.

- B7SCs, passed left-justified and space-filled in as many as necessary of the WA1 input fields B10SC-1, B10SC-2 and B10SC-3.

- B10SCs, passed in as many as necessary of the WA1 input fields B10SC-1, B10SC-2 and B10SC-3.

For functions that involve multiple input streets, the input streets specified in a call must all be in the same form, either all street names, or all PB5SCs (MSW only), or all B5SCs, or all B7SCs, or all B10SCs. Note that the first byte of all input street code fields is a borough code. When input streets are specified using street code input, the contents of the separate WA1 input borough code field (MSW only) is ignored.

Local street name validation (see Section IV.5) is not performed when the street input is in the form of five-digit street codes, but it is performed with seven-digit and ten-digit street code input.

NAPs and Street Code Input

With respect to a NAP of a simplex, as with conventional street input, the user has the option to specify the input datum either in the form of the name (in this case, the NAP) or its B5SC. However, five-digit street code input is prohibited for a NAP of a complex or a constituent entity of a complex (it is rejected with a GRC value of '07'); instead, 7-digit or 10-digit street code input (B7SC or B10SC) is accepted. The reason for this restriction is that different entities of the same complex may be located within different block faces, tax lots, census blocks, administrative or political districts etc., so that the B5SC may not be specific enough to enable Geosupport to determine the proper set of output data to return.

IV.9 Recapitulation

This section recapitulates the discussion of street codes. A B5SC value represents a New York City street (or a pseudo-street, non-street feature, place name or intersection name) and is assigned to all of the street's aliases, that is, to all of the names by which that street, or any part of it, is known to Geosupport. Therefore, the B5SC is a suitable item to use as an access key for street-related geographic retrieval, since then retrieval will be consistent with respect to street name aliases. That is, retrieval will succeed regardless of which alias is used at the time of record creation and which is used at retrieval time.

A B10SC value represents a particular spelling of a particular name for a street (along with all of that name's partial street names, if any). Two B10SC values are identical in their first six bytes (the B5SC values) if and only if the street names to which those B10SC values correspond are aliases (names for the same street). Since B10SC values correspond uniquely to a single spelling of a single name for a street, they can be used to automatically resynchronize the B5SCs stored in an application file to reflect street code assignment changes made in new Geosupport releases. Such resynchronization is essential, and is the user's responsibility. A Street Name/Street Code Change Bulletin, and a Geosupport file called the Street Code Change File (SCCF), are made available in each new Geosupport release to facilitate user-developed procedures for street code resynchronization. If an application file contains stored B10SCs, the SCCF can be used to develop a fully automated street code resynchronization procedure that directly accesses only those records in the application file containing B10SC values that must be updated, and that updates those B10SCs; this is the optimal resynchronization method. If street names and B5SCs are stored in the file, but not B10SCs, then the B5SCs can be resynchronized by using Function 1N and processing every record in the file. If only B5SCs are stored in the file, then the change bulletin must be used, and records containing B5SCs involved in changes must be individually researched.

Some streets in the city have certain names that are only locally valid, that is, valid only for a portion of the street. The set of all names for a given street is partitioned into 'local street name groups' corresponding to portions of the street where various street names are valid. Two names for a street are in the same local group if and only if they are valid for exactly the same portion of the street. Each local group is assigned a Local Group Code (LGC) value, which is a number from '01' to '99' that labels the group relative to all of the local groups for that street. The B5SC value and LGC value are concatenated to form the B7SC value, which is assigned to every name belonging to the corresponding local group, and only to those names. Each local group has a single member designated as the principal street name for that group. Each portion of a street has one local group designated as the preferred local group for that portion. The principal name of the preferred local group is called the preferred street name for that portion of the street. The preferred street name is obtainable for any specific location on a street. For example, for an address, the preferred street name is obtained by calling Function 1 to obtain both the B5SC and the preferred LGC, concatenating these to form a B7SC, and using the latter as input to a call to Function DG. For an intersection, street segment or block face, Function 2, 3 or 3C is called, respectively, instead of Function 1.

The B5SC, B7SC and B10SC can be viewed as forming a hierarchy in which the greater the length of the item, the more restricted the set of street names represented. The B5SC represents all the names for the street. The B7SC represents all the names that are valid for a particular portion (possibly all) of the street. The B10SC represents a particular name (and any unambiguous partial street names generated from it).

The methodology that is used to assign street codes to the Non-addressable Place Names (NAPs) of a complex and its constituent entities is analogous to the methodology used with street names. The B5SC represents all the NAPs of both the complex as a whole and all of its constituent entities. A distinct B7SC represents all the NAPs that are valid for a particular portion (possibly all) of the complex, that is, all the NAPs that are valid either for the complex as a whole or for a particular constituent entity. A distinct B10SC represents each individual normalized spelling of a specific NAP (and any unambiguous partial names generated from it).

For the reader's convenience, two reference tables summarizing street codes are below. Table IV-1 is a summary of the various street code items used by Geosupport, conventional abbreviations for them, and their lengths in bytes. The abbreviations listed in Table IV-1 are used throughout the remainder of the UPG. In these abbreviations, 'B' represents the standard Geosupport one-byte Borough Code, as described in Appendix 3, and 'P' means that the item is packed. Note: Packed applies to MSW only. Table IV-2 is a summary of the three main types of street code items, indicating the display function that accepts each as input, and what street name that display function returns as output. Table IV-2 is written to describe the assignment of street codes to street names, but it applies analogously as well to the NAPs of a complex and its constituent entities.

Table IV-1: Notation for Street Code Items

Item Abbreviation	Item Name	Length (Bytes)
5SC	Five-digit Street Code	5
P5SC (MSW only)	Packed Five-digit Street Code	3
B5SC	Borough and Five-digit Street Code	6
PB5SC (MSW only)	Packed Borough and Five-digit Street Code	4
7SC	Seven-digit Street Code	7
B7SC	Borough and Seven-digit Street Code	8
10SC	Ten-digit Street Code	10
B10SC	Borough and Ten-digit Street Code	11
LGC	Local Group Code (6th and 7th digits of 10SC)	2
SNC	Street Name Code (8th, 9th and 10th digits of 10SC)	3

5SC + LGC = 7SC

B5SC + LGC = B7SC

B5SC + LGC + SNC = B7SC + SNC = B10SC

Table IV-2: Summary of Street Code Items

Type of Street Code	Geography Represented	Corresponding Street Name(s)	Applicable Display Function and Its Output Datum
B5SC	A street	All names valid for all or any portion of the given street	D - returns primary name
B7SC	The portion (possibly all) of a street where a group of names is valid	All the names in the given local street name group	DG - returns principal name of local group
B10SC	The portion (possibly all) of a street where a specific name is valid	One spelling of one name (and any unambiguous partial names generated from it)	DN - returns the unique name to which the given B10SC corresponds

CHAPTER V: ADDRESS PROCESSING - FUNCTIONS 1, 1A, 1E

V.1 Introduction

This chapter and the following two chapters discuss in detail the various types of geographic locations that Geosupport can accept as input, and the Geosupport functions that process them. This chapter, Chapter V, discusses addresses and Functions 1, 1A and 1E. Chapter VI discusses properties (tax lots) and buildings and Functions 1A (covering aspects not discussed in Chapter V), BL and BN. Chapter VII discusses 'street configurations' (geographic locations that are specified in terms of combinations of streets) and Functions 2, 3, 3C and 3S.

The following topics are discussed in this chapter:

- (Section V.2) House number normalizing, and Geosupport's normalized house number formats: the HNI (MSW only), HNS (COW only), and HND.

- (Section V.3) The two types of input data accepted by the address-processing functions: addresses and Non-Addressable Place Names (NAPs). The two ways of specifying an input address: in 'parsed form' (passing the address number and street in separate fields) and in 'free form' (passing them together in a single field).

- (Section V.4) Distinctions among the address-processing functions with respect to which addresses are accepted and which rejected, and with respect to the validation significance of acceptance and rejection.

- (Section V.5) The output data returned by each address-processing function.

- (Section V.6) A special feature for processing duplicate addresses.

- (Section V.7) A special feature for processing Marble Hill and Rikers Island locations. (This feature is implemented not only for the address-processing functions, but also for some of the street configuration functions.)

- (Section V.8) A special feature for processing addresses on Ruby Street, a street along the Brooklyn-Queens border.

- (Section V.9) A special feature for processing 'vanity addresses'. In such an address, the street name refers to a street other than the one upon which the appurtenant building entrance is actually located.

- (Section V.10) A special feature for processing out-of-sequence addresses.

V.2 Address Numbers ('House' Numbers), Normalization and Formats: HNI, HNS and HND

Address numbers identify buildings along streets, and are combined with street names and addressable place names (see Section III.6) or with street codes (as surrogates of street names or place names) to form addresses. Address numbers are commonly called 'house' numbers (although this term is a misnomer, since many addresses refer to buildings other than houses). To be consistent with common parlance and with other Geosupport documentation, the term 'house number' will be used instead of 'address number' in the remainder of this document, except in literal citations of Geosupport reject messages, since those messages use the term 'address number'.

Applications can pass a house number to any of the address-processing functions in character form, in the 12-byte WA1 input House Number field for MSW and the 16-byte WA1 input House Number field for COW. A house number passed in this manner need not be in any particular format, but could be a 'raw', un-formatted house number. Alternatively, house numbers can be passed in a 6-byte WA1 input field in a special Geosupport format called the House Number in Internal format (HNI), which presumes, the application will have obtained the HNI from a previous Geosupport call.) HNIs are only used with MSW. A new Geosupport format called House Number in Sort Sequence (HNS) is used for COWs.

When a house number is passed to Geosupport in the 12-byte WA1 for MSW or the 16-byte WA1 for COW input House Number field, Geosupport normalizes it. The house number normalization algorithm is complex, and a full description of it is beyond the scope of this document, but some aspects are discussed below. If normalization is successful, an output normalized house number is produced in two standard formats, the 12-byte or 16-byte output House Number in Display format (HND) and the 6-byte output House Number in Internal format (HNI) or the 11-byte House Number in Sort format (HNS), and both of these are returned to the application in WA1. The HND is in character form and is suitable for display, for example, on application screens, reports and mailing labels. While the HNS format contains character data, it is intended for Geosupport internnal use. To conserve space, users may store this value in their files.
The HNI format contains packed decimal data, and is the format that Geosupport uses internally to perform its address-matching routines. The HNI is not documented in detail herein, and is of little direct relevance to most users. However, to conserve disk space in application files in which house numbers must be stored in some form, users can store the 6-byte HNI in their files rather than the 12-byte HND for MSW or the 11-byte HNS in their files rather than the 16-byte HND for COW, and then use any of the display functions, Functions D, DG and DN, to obtain the house number in HND format for display, as described below.

Processing of HNIs or HNSs by the Display Functions

The processing of an input HNI or HNS by a display function consists only of forming and outputting the HND. The successful processing of an input HNI or HNS by a display function implies that the HNI or HNS conforms to Geosupport's format requirements for HNIs or HNSs, but does not imply that the HNI or the HNS forms part of a valid address.

The display functions can process up to two input HNI or HNS values in a single call, using the two input HNI or HNS fields and two output HND fields in WA1. If two input HNIs or HNSs are supplied, they are processed independently of each other and are not treated as forming an address range. If only one input HNI or HNS is supplied, it may be passed in either of the input HNI or HNS fields.

The display functions return one output HND for each validly formatted input HNI or HNS. For each input HNI or HNS that is invalid, the display functions return all question marks (the character '?') in the corresponding output HND field. In addition, if at least one input HNI or HNS is invalid, the GRC value '13', Reason Code value '9' and corresponding Message are issued.

The display functions can also be used to obtain street names corresponding to input street codes. (The processing of street codes by the display functions is discussed in detail in Section IV.6.) In a single call, the display functions can process input HNIs or HNSs without input street codes, input street codes without HNIs or HNSs or both types of input. If both HNIs or HNSs and street codes are provided as input data to a display function call, they are processed independently of each other and are not treated as forming an address. In particular, the display functions perform no address validation.

HNIs or HNSs as Input to the Address-Processing Functions

The user has the option of providing input house numbers to the address-processing functions in the form of an HNI or HNS instead of a ‘raw’ unprocessed house number. This feature is useful for processing an application file that already contains house numbers in HNI or HNS format from a previous pass through Geosupport. The use of this feature slightly improves execution efficiency by allowing Geosupport to circumvent the house number normalization routine.

House Number Format Standards

'Raw' (un-normalized) input house numbers must conform to certain Geosupport standards, which are based on the characteristics of New York City’s addresses. If an input house number does not satisfy these standards, Geosupport is unable to normalize it and rejects the call. The house number standards include the following, among others:

- Conformance to a set of allowable characters
- A limitation on the total length of the ‘basic house number’ (this term and the term ‘house number suffix’ are defined below)
- Limitations on the number of digits and maximum numeric values of the basic house number, if it does not contain a hyphen; or such limitations on the portions of the basic house number preceding and following the hyphen, if a hyphen is present
- Validity of the house number suffix (discussed below), if one is present

Every valid New York City house number conforms to the above standards.

The ability of Geosupport to normalize an input house number does not by itself signify that that house number, together with the input borough and street, form in combination a valid New York City address. Successful normalization signifies only that the input house number conforms to Geosupport’s house number format criteria. Only the successful completion of a two-work-area call to one of the address-processing functions has significance with respect to the geographic validity of the input address. (See Section II.4 for a discussion of the distinction between the validations performed by one- and two-work-area calls.)

New York City house numbers consist of a ‘basic house number’, possibly followed by a ‘house number suffix’. (Note: the basic house number and house number suffix are not to be confused with the digits to the left and right of the hyphen in a hyphenated house number. For example, in the Queens address ‘240-55 1/3 DEPEW AVENUE’, ‘240-55’ is the basic house number, and is hyphenated; ‘1/3’ is the house number suffix.) Only a very small percentage of New York City addresses have house number suffixes. The following are some examples of valid New York City addresses containing house number suffixes (highlighted in bold type):

519 **Front** East 12th Street (Manhattan)
625 **Rear** Smith Street (Brooklyn)
120 **1/2** First Avenue (Manhattan)
240-55 **1/3** Depew Avenue (Queens)

469 **1/4** Father Capodanno Boulevard (Staten Island)
470 **A** West 43rd Street (Manhattan)
171**C** Auburn Avenue (Staten Island)
20-29 **Garage** 120th Street (Queens)

Input basic house numbers may contain a dash character (the character '-'), which can serve either as a hyphen, as with most house numbers in Queens and some house numbers in other boroughs, or as a range separator character.

- House Number Ranges: Addresses in New York City are often expressed in ranges, using a dash to separate the low and high house numbers of the range. For example, 22-28 Reade Street in Manhattan represents the range of even addresses consisting of 22 Reade Street, 24 Reade Street, 26 Reade Street and 28 Reade Street, all of which are valid individual addresses for the same building. In other words, in this example, the character string '22-28' is not an individual house number, but represents a range of house numbers, in which the dash serves as a range separator character, and the number to the left of the dash, 22, as well as that to the right of the dash, 28, constitute by themselves valid individual house numbers for Reade Street.

- Hyphenated House Numbers: Consider the Queens address 22-28 36th Street. The house number portion of the address, 22-28, consists of the same character string as the above Reade Street example, but it has a very different meaning in the two cases. In the Reade Street case, 22-28 represents a range of even house numbers; in the 36th Street case, 22-28 is a single hyphenated house number, not a range of several unhyphenated house numbers. In a hyphenated house number, the digits to the left and to the right of the hyphen in combination form a single house number; the digits on one side of the hyphen are not by themselves geographically meaningful. For example, 22 36th Street and 28 36th Street are not valid Queens addresses. In addition, the position of the hyphen within a hyphenated house number is significant. For example, consider the addresses 13-103 41st Avenue and 131-03 41st Avenue. These are two distinct addresses on the same Queens street, even though the house numbers consist of the same sequence of digits and differ only in the position of the hyphen.

Geosupport's house number normalization algorithm interprets a dash encountered in an input house number either as a hyphen or as a range separation character, depending on the borough, the street (some streets do not conform to the norm for their borough with respect to house number hyphenation) and other criteria.

- When Geosupport interprets the dash as a range separation character: In normalizing the input house number, both the dash itself and the portion of the basic house number to the right of the dash are deleted. As one consequence of this, when the input to a two-work-area call is an address range, only the address formed from the house number to the left of the dash is validated; the house number to the right of the dash is ignored and no conclusion can be drawn about its validity from the success or failure of the call. For example, 22-28 Reade St in Manhattan is normalized as 22 READE STREET; the '28' is ignored during normalization, and is not validated as an individual house number in a two-work-area call.

- When Geosupport interprets the dash as a hyphen: In normalizing the input house number, the digits on both sides of the hyphen are retained, as is the hyphen itself.

If Geosupport determines that an input house number in character form has a missing or inappropriately present dash, then whenever it is feasible, Geosupport modifies the house number to correct the error before normalizing it. (Geosupport never modifies input HNIs or HNSs.) Geosupport will make such a modification automatically (without user request), but only if the intended address is clear and unambiguous and is valid for the function being called, and a valid address could not be formed by normalizing the input house number in a different fashion. Two types of such dash-related modifications are as follows:

- When an input house number does not contain a dash, but Geosupport determines that the house number should be hyphenated: Geosupport inserts a hyphen, provided it can determine the proper position of the hyphen unambiguously so that a valid address results. For example, the input address 6603 Booth Street in Queens is normalized as 66-03 BOOTH STREET; the input address 63101 Alderton Street in Queens is normalized as 63-101 ALDERTON STREET.

- When an input house number contains a dash, but Geosupport determines that the presence of the dash is erroneous (i.e., the house number is invalid whether the dash is interpreted as a hyphen or as a range separator): Geosupport concatenates the digits to the left and right of the dash without retaining the dash itself, provided that this results in a valid address. For example, 10-22 38th Street in Brooklyn is normalized as 1022 38 STREET.

Whenever the house number normalizer makes an assumption about, or a dash-related modification to, an input house number, Geosupport so informs the calling application by issuing a warning condition. A warning is issued, for example, when Geosupport assumes that an input dash is a range separator and then normalizes the house number by deleting the dash and digits following it, or when it assumes that a required hyphen is missing and inserts one.

When Geosupport is unable to normalize an input house number without making a dash-related modification so that a valid address results, and there is more than one type of dash-related modification that would result in a valid address, the input is considered ambiguous. For such a rejection, the Message would list the possible valid forms of the input address. This assists the user to determine how the input house number should be modified to make it valid. For example, consider the input 10-14 Lexington Avenue in Manhattan. Lexington Avenue has unhyphenated addresses only. There are two reasonable interpretations of the user's intended input in this example. These are 10 Lexington Avenue, which assumes the input is an address range, and 1014 Lexington Avenue, which assumes the dash is an inappropriately present hyphen. All of the address-processing functions consider both of these to be valid addresses. Initially, 10-14 Lexington Avenue in Manhattan was rejected as ambiguous, but, at user request, the first successful house number is accepted; i.e. 10 Lexington Avenue in Manhattan.

In the borough of Queens, the great majority of streets have hyphenated house numbers only; a few streets have unhyphenated house numbers only, and a few streets have 'mixed hyphenation', that is, both hyphenated and unhyphenated house numbers. In the other four boroughs, all but a few streets have unhyphenated house numbers only, a few streets have hyphenated house numbers only, and a few streets have mixed hyphenation. Riverside Drive in Manhattan is an example of a mixed-hyphenation street. A small stretch of Riverside Drive running north from West 156th Street has hyphenated even addresses ranging from 156-00 to 159-34 (with some gaps). The remainder of Riverside Drive has unhyphenated addresses only.

Information on the address hyphenation status of each of the city's streets is maintained internally within Geosupport. The house number normalizer makes use of this information when analyzing an input house number that contains a dash character. Dash analysis is particularly complex for mixed-hyphenation streets, for which a dash could be either a hyphen or a range separator. For example, 156-158 Riverside Drive is a valid range of unhyphenated addresses assigned to a building located near West 88th Street, while 156-10 Riverside Drive is a valid single hyphenated address assigned to a building located near West 156th Street.

V.3 Specifying Input Data to the Address-Processing Functions: NAPs, Parsed-Form Addresses and Free-Form Addresses

Functions 1, 1A and 1E are Geosupport's address-processing functions. They accept as input both conventional street addresses and certain Non-Addressable Place Names (NAPs) (described in Section III.6 and further discussed in Section IV.6).

- Addresses can be specified in parsed form, that is, with the house number and street specified in separate WA1 input fields. The street can be specified either as a street name or a street code. Non-street feature names, pseudo-street names and intersection names may not be used. There are two options for specifying an address in parsed form:

    - Parsed-form addresses using street name: Specify a borough code, street name and house number (using the WA1 input borough code and street name-1 fields and either the 12-byte or 16-byte WA1 input house number field or the 6-byte WA1 input HNI field or the 11-byte HNS field).

    - Parsed-form addresses using street code: Specify a borough code, street code and house number. The borough code and street code may be specified using any of the WA1 input combined borough code and street code fields (the B5SC, PB5SC, B7SC or B10SC) for MSW. For COW, there is only one borough code field. The house number may be specified using either the 12-byte or 16-byte WA1 input house number field or the 6-byte WA1 input HNI field or the 11-byte HNS field.

- Alternatively, addresses can be specified in free form, that is, with the house number specified together with the street name in the WA1 input Street Name field, as described below. Non-street feature names, pseudo-street names, intersection names and partial street names (see Section III.4) may not be used. When addresses are specified in free form, the input house number and HNI or HNS fields are not used.

- NAPs are specified in the same fashion as addresses, as described above, except that no house number is supplied. (If a house number is supplied with a NAP, Geosupport ignores the house number and issues a warning.)

Free-form addresses are addresses in which the house number and street name are stored together in a single field, as they might appear in the address line of a mailing address. When an application passes all blanks in the WA1 input house number and HNI or HNS fields, and Geosupport determines that the WA1 input Street Name field does not contain a NAP, Geosupport assumes that the latter contains a free-form address, and attempts to parse the contents into a house number followed by a street name.

Since both house numbers and street names vary in length, and may be separated by a varying number of blanks, these items will not be in predictable positions within a free-form address. Therefore, when processing a free-form address, Geosupport must parse the contents of the input street name field to attempt to identify and logically separate the house number and the street name. If this is successful, the processing proceeds as with parsed-form address input. If an input free-form address contains any extraneous data following the house number and street name, such as an apartment number, neighborhood name, borough name or zip code, Geosupport attempts to recognize those data as extraneous, in which case it ignores them.

Geosupport's processing of free-form addresses is complex and is not as reliable as that for parsed-form address input. It is strongly recommended that, whenever possible, applications be designed to pass input addresses to Geosupport in parsed form, that is, to pass input house numbers and input street names in separate fields.

V.4 Input Address Acceptance/Rejection and its Validation Significance

The address-processing functions differ significantly among themselves with respect to which input addresses they accept and reject, and with respect to the significance of the validation of an input address implied by acceptance or rejection. These distinctions are discussed below.

Function 1. Function 1 accepts an input address if and only if it falls within the administrative address range allocated to some block face (described below). Thus, Function 1's acceptance of an input address does not by itself validate whether the input address is the actual address of a building, but only whether it falls within an administrative address range.

The administrative address range allocated to a block face is the set of addresses that actually are, or potentially may be, assigned to buildings on that block face. Administrative address ranges are allocated to block faces by the offices of the Borough Presidents. In many cases, the administrative address range allocated to a block face is broader than its current 'actual' address range (i.e., the range encompassed by the lowest and highest actual addresses of existing buildings on the block face). This reserves addresses for new buildings that might be built on that block face in the future. (To 'shoehorn' new buildings between existing buildings, it is sometimes necessary to assign house numbers with suffixes like 1/2 and 1/3.)

In theory, an administrative address range encompasses all of the actual addresses of existing buildings on the block face. However, there are discrepancies from this in reality for a relatively small number of block faces, as well as temporary discrepancies caused by Geosupport data errors.

An administrative address range may also encompass nonexistent addresses, either between the low and high actual addresses of the block face or beyond them. For example, consider the block face on the east side of East 28 Street between Avenues I and J in Brooklyn. The administrative address range allocated to this block face is 901-999. Function 1 would accept any odd address between 901 and 999 on E 28 Street in Brooklyn as input, whether or not that input address is a valid address of an existing building. In reality, the lowest and highest actual house numbers of existing buildings on this block face (as of the writing of this document) are 901 and 985, and within this range there are gaps in actual addresses. For example, there are buildings on East 28 Street with the house numbers 925 and 929, but there is not currently a building with the house number 927, nor are there buildings with any of the odd house numbers from 987 through 999. Nevertheless, all of these house numbers will result in successful Function 1 calls, since they all fall within the administrative address range.

Function 1E. With the exception of a rare case discussed below, Functions 1 and 1E accept the same addresses and reject the same addresses, and the validation significance of acceptance and rejection is the same for both functions.

The exceptional case is that of an address that is split among more than one Election District (ED). As of this writing, there is only one instance of this case, 3333 Broadway in Manhattan; it is split among three EDs. Therefore, for this address, Function 1E is unable to determine an ED (or any of the higher-level political districts). Since the primary purpose of Function 1E is to provide the political geography for an address, Function 1E rejects this address with a GRC value of '56'. However, portions of this building in specific EDs can be identified using house number suffixes, 'A' through 'E': 3333A through 3333C Broadway are in ED 94 of Assembly District (AD) 70; 3333D Broadway is in ED 82 of AD 70; and 3333E

Broadway is in ED 83 of AD 70. Function 1E accepts these addresses as input, and returns the political districts specific to the input. Functions 1 and 1A accept both the un-suffixed and suffixed addresses.

Function 1A. Function 1A accepts an input address if and only if the address falls within one of the following two cases:

- Valid actual address. If the input address is a valid address of an existing building on a property, there is a normal completion (Geosupport Return Code = '00').

- Pseudo-address. If the input address is a 'pseudo-address' , a warning is issued (GRC = '01', Reason Code = '8' or '9'). Pseudo-addresses (not to be confused with pseudo-street names) are discussed in Section VI.5.

If the input address is neither a valid address of an existing building nor a pseudo-address, Function 1A rejects that input address. This is true even if the input address falls within an administrative address range allocated to a block face and is therefore accepted by Functions 1 and 1E. Thus, Function 1A's criterion for accepting an input address is more stringent than those of Functions 1 and 1E, and the validation significance of acceptance differs accordingly.

V.5 Output Data Returned in Work Area 2

The address-processing functions differ significantly with respect to the output data they return.

Function 1. Function 1, when called using two work areas, performs block face-level processing. Almost all of the items that Function 1 returns in WA2 are associated with the entire block face, and do not vary with the specific input address within that block face. Among these items is a set of geographic district identifiers, such as Census Tract and Block, Police Precinct and Community District.

One piece of information returned by Function 1 that does vary with the specific input address is a pair of spatial coordinates. This identifies the approximate location of the given address on the earth's surface. (Note: spatial coordinate values are not returned if the address lies on an irregularly curved street segment (i.e., a curve that is not an arc of a circle); in that case, the WA2 field called the Curve Flag contains the value 'I', and a warning is issued with Reason Code 'P'. See the discussions of the Curve Flag and Spatial Coordinates in Appendix 3.)

Community School District (CSD) boundaries split some block faces, and in those cases, Function 1 returns the CSD value that is appropriate for the specific input address. However, the high and low house numbers returned in WA2 always correspond to the entire block face, not to the portion of the block face within the given CSD. When the block face is split by a CSD boundary, Function 1 issues a warning (with Reason Code 'E'), indicating that the CSD value that has been returned does not apply to the entire address range that has been returned.

An example of a block face that is split by a CSD boundary is the odd-address side of FARRAGUT ROAD in Brooklyn between EAST 105 STREET and EAST 108 STREET. The address range for the entire block face is 10501 to 10799. The subrange from 10501 to 10599 is in CSD 18; and the subrange from 10601 to 10799 is in CSD 19. If the input to a Function 1 call is 10559 FARRAGUT ROAD, '18' is returned in the WA2 CSD field, but 10501 and 10799 are returned as the address range for the block face. In addition, a warning is issued with a GRC value of '01', a Reason Code value of 'E' and an appropriate Message.

The information that Function 1 returns in WA2 also includes two lists of street codes for the cross streets at both ends of the block face. Applications can use these cross streets to identify address-based data to block faces or street segments. In many applications, the consolidation of data for individual locations to the level of the block face or street segment can significantly improve the efficiency of a municipal operation. The conversion of address-based data to segment-based data is further discussed in Section VII.3. If the application has a need to display the street names of the cross streets, the Cross Street Names Flag in WA1 can be turned 'on' and the names will be returned in the List of Street Names in WA1 (see entries for Cross Street Names Flag and List of Street Names in Appendix 3). Note that the cross street names feature incurs processing overhead, and should only be used when necessary.

The long WA2 option is available for Functions 1 and 1E. The additional data provided by the long WA2 option includes the LION Segment ID.

Functions 1 and 1E have been enhanced to allow a user to receive <u>roadbed-specific information</u> in place of information based upon the center line of a multi-roadbed street. A user requests roadbed-specific information via the "Roadbed Request Switch". This means that a Function 1 or 1E call with this switch set will return the roadbed-specific geocodes, assuming that the input street has multiple roadbeds. Examples of geocodes that would be different include Segment ID, Segment Type Code, X-Y coordinates, LION Key and possibly cross streets. An additional file has been added to the Geosupport system to handle this data. Users who prefer non-roadbed-specific information, which assumes a single roadbed for all roads, are not required to make any changes.

<u>Function 1E.</u> Function 1E, when called using two work areas, returns all of the WA2 data items that Function 1 returns. In addition, Function 1E returns the following political district identifiers in WA2: Election District, State Assembly and Senate Districts, City Council District, Congressional District and Municipal Court District. To obtain these additional items, Function 1E accesses not only the foreground files accessed by Function 1, but also additional files. Because of this additional execution overhead, <u>it is advisable for users to design their applications to call Function 1 rather than Function 1E, unless there is a specific need for the additional political district information that Function 1E provides.</u>

Function 1E handles cases where a Community School District boundary splits a block face in the same manner as Function 1 does. In addition, Election District boundaries can also split block faces, and Function 1E handles those cases similarly. This includes the special case of the addresses 3333A through 3333E Broadway discussed in Section V.6.

As indicated before, Function 1E permits roadbed-specific information to be returned to the user. For more information, refer to the description as part of Function 1.

<u>Function 1A.</u> Function 1A, when called using two work areas, performs property (i.e., tax lot)-and building-level processing. Function 1A returns information in WA2 associated with the specific property and building (if any) containing the input address. This information includes the property identifiers (tax block and tax lot numbers), and a list of all addresses of all buildings on the property (or as many as will fit in WA2). Function 1A's output information is discussed in detail in Section VI.4.

V.6 Duplicate Addresses

New York City has a small number of <u>duplicate addresses</u>, which are not data errors in Geosupport files, but real duplicates in the assignment of house numbers to buildings or in the allocation of administrative address ranges to block faces. Some duplicate addresses were created when formerly independent towns were consolidated into one of the city's boroughs. Other duplicates involve situations in which a developer or other entity has given the same name to a private street that the city has given to a public street in the same borough. When two streets in the same borough have the same name, Geosupport treats them as portions of a single street, regardless of how far apart they are geographically.

Geosupport's address-processing functions, Functions 1, 1A and 1E, when called using two work areas, have a special feature to process duplicate addresses. This feature involves the use of Duplicate Address Pseudo-Street Names (DAPSs), described in Section III.6. DAPSs are assigned to every street that has duplicate addresses, providing a means for users to specify unambiguously a particular instance of such an address.

Whether a given input address is processed as a duplicate address depends on the function. Functions 1 and 1E behave identically with respect to duplicate address processing, both in the set of addresses they consider to be duplicates, and in the way addresses are processed. However, Function 1A differs from Functions 1/1E in both of these respects.

There are two cases of duplicate address situations:

- Case 1: Overlap of Administrative Address Ranges Allocated to Two Block Faces: The same administrative address range (discussed in Section V.4) or portion thereof is allocated to two different block faces on the same street in the same borough. That is, an administrative address range allocated to one block face along a street contains, coincides with, or otherwise overlaps with, that allocated to another block face along the same street. Any input address that is within such an overlap is processed as a duplicate address by Functions 1 and 1E.

- Case 2: Duplication of an Address or Address Range Assigned to Two Buildings: The same address or range of addresses is assigned to two different buildings on the same street in the same borough. Any such input address is processed as a duplicate address by Function 1A.

In general, an address that is an instance of Case 2 is also an instance of Case 1, but the reverse is not necessarily true. That is, almost all addresses that are processed as duplicates by Function 1A are also processed as duplicates by Functions 1/1E, but there are many addresses that are processed as duplicates by Functions 1/1E but are not processed as duplicates by Function 1A.

Hillside Avenue in Queens has addresses that exemplify both of the above cases. A Case 1 example is the following. There is a block face of Hillside Avenue in the Bellerose neighborhood of Queens to which the administrative address range 239-02 to 239-10 is allocated. There is another block face of Hillside Avenue, in the Douglaston neighborhood of Queens, to which the administrative address range 239-02 to 239-20 is allocated. Since the Bellerose range is entirely contained within the Douglaston range, Functions 1 and 1E process every house number in the Bellerose range (all the even house numbers on Hillside Avenue from 239-02 to 239-10) as a duplicate address.

Despite the administrative address range duplication, not every individual even house number on Hillside Avenue between 239-02 and 239-10 is a valid actual building address in both Bellerose and Douglaston, i.e., is an instance of Case 2. Function 1A does not process an address as a duplicate unless it is assigned to two different buildings, even if Functions 1 and 1E process that address as a duplicate.

Function 1A faces three possibilities when processing an address that Functions 1/1E consider to be a duplicate, as illustrated by the following Hillside Avenue examples:

- 239-02 Hillside Avenue is assigned to buildings in both Bellerose and Douglaston. Therefore, this address is processed as a duplicate address by Function 1A, as well as by Functions 1/1E.

- 239-06 Hillside Avenue is assigned to only one building, in Bellerose. Therefore, this address is processed normally, not as a duplicate address, by Function 1A, even though it is processed as a duplicate address by Functions 1/1E.

- There are no buildings to which 239-04 Hillside Avenue is assigned. Therefore, this address is rejected as entirely invalid by Function 1A, even though it is processed as a duplicate address by Functions 1/1E.

<u>Processing of a duplicate address.</u> When an address-processing function considers an input address to be a duplicate, it processes that address as follows.

- <u>If the street in the input address is specified using the conventional street name, the call is rejected</u> with a Geosupport Return Code of '75'. The accompanying Message informs the user that the function considers this input address to be a duplicate address, and indicates the two DAPSs that could be used to specify this address unambiguously. For example, 239-02 Hillside Avenue is considered a duplicate address by all of the address-processing functions, so all of them would reject the input 239-02 HILLSIDE AVENUE and would return a GRC of '75' and the Message:

    DUPLICATE ADDRESS-USE HILLSIDE AVENUE BELLEROSE
    OR HILLSIDE AVENUE DOUGLASTON

- <u>If the user specifies the input address using one of the DAPSs instead of the conventional street name, it is accepted</u>. For example, 239-02 HILLSIDE AVENUE BELLEROSE is accepted by all of the address-processing functions, as is 239-02 HILLSIDE AVENUE DOUGLASTON. These functions return output information that is specific to the block face (Functions 1/1E) or the tax lot and building (Function 1A) pinpointed by the DAPS.

To summarize, <u>for an address that the function being called considers to be a duplicate, the conventional street name is rejected; only a DAPS is accepted as an input street name for such addresses.</u>

<u>Processing of a **non**-duplicate address on a street that has DAPSs.</u> Now suppose that a particular address-processing function considers a given input address to be a valid <u>non</u>-duplicate. If the street does not have DAPSs (i.e., if the street does not have any addresses that are considered to be duplicates by any of the address-processing functions), then the processing does not involve the duplicate address processing feature. If the street does have DAPSs, then the address is processed as follows.

- <u>If the address is specified using the conventional street name</u>, it is accepted. For example, 239-20 Hillside Avenue is considered a non-duplicate address by all of the address-processing functions. Therefore, they all accept 239-20 HILLSIDE AVENUE as input. 239-06 Hillside Avenue is considered a non-duplicate address by Function 1A (it is valid for one building only, which happens to be in Bellerose), but it is considered a duplicate address by Functions 1/1E. Therefore, Function 1A accepts 239-06 HILLSIDE AVENUE as input, but Functions 1/1E reject this as a duplicate address for which DAPS input is required.

- <u>If the address is specified using DAPSs</u>, the processing depends on the function:

- Functions 1/1E reject the address. For example, Functions 1/1E reject 239-20 HILLSIDE AVENUE BELLEROSE, as well as 239-20 HILLSIDE AVENUE DOUGLASTON. Functions 1/1E treat DAPSs as valid only for addresses they consider to be duplicates, and reject DAPSs for addresses that these functions do not consider to be duplicates.

- Function 1A's processing depends on whether Functions 1/1E consider the address to be a duplicate:

    - If Functions 1/1E consider the address to be a non-duplicate, Function 1A rejects the address. For example, Function 1A rejects the input 239-20 HILLSIDE AVENUE BELLEROSE, as well as 239-20 HILLSIDE AVENUE DOUGLASTON.

    - If Functions 1/1E consider the address to be a duplicate, Function 1A accepts the address when it is specified using the DAPS corresponding to the neighborhood where the address is valid, and rejects the address when it is specified using the other DAPS. For example, as mentioned above, 239-06 Hillside Avenue is valid only for a building in Bellerose. Therefore, Function 1A accepts 239-06 HILLSIDE AVENUE BELLEROSE, but it rejects 239-06 HILLSIDE AVENUE DOUGLASTON. Note that this contrasts with the behavior of Functions 1/1E, which reject both DAPSs when they consider an address to be a non-duplicate.

We now summarize the duplicate address processing feature. Functions 1 and 1E behave identically with respect to duplicate address processing: they consider the same addresses to be duplicates, and they process all addresses in exactly the same way, both those they consider to be duplicates and those they do not. However, Functions 1/1E differ from Function 1A in certain respects.

- Functions 1/1E differ from Function 1A in which addresses they consider to be duplicates. Almost all addresses that Function 1A considers to be duplicates are also considered duplicates by Functions 1/1E. However, there are many addresses that Functions 1/1E consider to be duplicates that Function 1A does not.

- If an address-processing function considers an input address to be a duplicate, the function rejects the conventional street name, and accepts DAPSs.

- If an address-processing function considers an input address to be a valid non-duplicate, it accepts the conventional street name.

- If an address-processing function considers an input address to be a valid non-duplicate, and the street is specified using a DAPS, the action taken depends on the function. Functions 1/1E reject both DAPSs, regardless of whether Function 1A considers the address to be a duplicate. Function 1A's action depends on whether Functions 1/1E consider the address to be a duplicate. If Functions 1/1E consider the address to be a non-duplicate, Function 1A rejects both DAPSs. If Functions 1/1E consider the address to be a duplicate, Function 1A accepts the DAPS that corresponds to the location where the input address is a valid address of a building, and rejects the other DAPS.

V.7 Marble Hill/Rikers Island

There are two New York City areas, Marble Hill and Rikers Island, that have the following idiosyncrasy:

each is more closely identified geographically with a borough other than the borough to which the area legally belongs. The former is referred to as the 'alternative borough'.

- Marble Hill: The legal borough is Manhattan, and the alternative borough is the Bronx. Marble Hill is located on the Bronx side (the north side) of a body of water separating Manhattan Island from the Bronx, and it has a land boundary with the Bronx but it is connected to Manhattan Island only by a bridge.

- Rikers Island: The legal borough is the Bronx, and the alternative borough is Queens. Rikers Island is physically connected to Queens via a bridge but is not connected to the Bronx.

Because of their locations, Marble Hill and Rikers Island receive many of their government services from their alternative borough, and therefore they are included in many operational and administrative districts of the alternative borough. For example, most of Marble Hill is in Bronx Community District (CD) 8, and the remainder of it is in Bronx CD 7. Marble Hill also has a Bronx zip code (10463). Rikers Island is assigned to Queens CD1.

In practice, both the legal borough and the alternative borough are used when specifying Marble Hill and Rikers Island locations. To accommodate this practice, Geosupport's address-processing and street configuration-processing functions have been designed to accept either the legal borough or the alternative borough as the input borough for Marble Hill and Rikers Island locations. For example, the Marble Hill address 150 WEST 225 STREET is accepted by the address-processing functions whether Manhattan or the Bronx is specified. The Rikers Island address 18-99 HAZEN STREET is accepted whether the Bronx or Queens is specified.

When an application makes a two-work-area call to any of the address-processing or street configuration-processing functions other than Function 3S, and specifies the alternative borough for a Marble Hill or Rikers Island location, Geosupport issues a warning with a Reason Code value of 'C' and an appropriate Message.

The information returned to the application in WA2 is the same regardless of which borough is specified as the input borough. However, the output borough name and street code(s) returned to the application in WA1 do depend on which borough is specified as the input borough. Each street in Marble Hill and Rikers Island has two street codes assigned to it, one for the legal borough and one for the alternative borough. The street code(s) and borough name that are returned in WA1 correspond to the input borough.

V.8 Special Ruby Street Processing

The address processing functions have a special feature to handle an anomaly involving a stretch of a street that lies along the Brooklyn-Queens border. On the Brooklyn side, this street is called Ruby Street; on the Queens side of the same physical street, it is called 75 Street. Many residents of the Brooklyn side of this street customarily specify Brooklyn as the borough but they specify the Queens street name, 75 Street, rather than the 'legal' Brooklyn street name, Ruby Street. A further complication is that there is a different Brooklyn street (in Bay Ridge, far from the Queens border) that is also called 75 Street. Geosupport handles these anomalies automatically, as follows.

When an address-processing function is called, and Brooklyn is specified as the input borough, and 75 STREET is specified as the input street name, Geosupport is able to determine from the input house number (assuming it is a valid house number) whether the address is on 75 Street in Bay Ridge or is actually on Ruby Street. If it is in Bay Ridge, it is processed normally. If it is on Ruby Street, then Geosupport takes the following actions:

- The street name RUBY STREET and the street code for Ruby Street are returned in WA1 instead of the normalized input street name, 75 STREET, and the latter's Brooklyn street code.

- WA2 is returned with a full complement of data.

- Geosupport issues a warning with a Reason Code value of either '6' or '7'. (Reason Code '6' indicates simply that the output street name and street code differ from the corresponding input values. Reason Code '7' indicates in addition that the input and output house numbers differ from each other in some way, as per Section V.2. See Appendix 4.)

V.9 Vanity Addresses

Vanity addresses are a type of geographically 'dislocated' address in which the street name is that of a different street from the one on which the building entrance is actually located. Developers sometimes use such addresses in the belief that a prestigious street name enhances the market value of a property.

1049 FIFTH AVENUE in Manhattan is an example of a vanity address. Although this is ostensibly a Fifth Avenue address, the building entrance to which this address is assigned is actually located on the south side of East 86th Street between Fifth and Madison Avenues. (In this case, the building has no frontages at all on the named street, although that is not a prerequisite to being a vanity address.)

Functions 1 and 1E process vanity addresses as follows. A warning with Reason Code 'V' is issued, along with a message that indicates the 'true' street name (the name of the street on which the building entrance is actually located). The output data returned, including cross streets and geographic district identifiers, pertain to the true block face. For example, the information returned for 1049 FIFTH AVENUE corresponds to the block face on the right (south) side of East 86th Street between Fifth and Madison Avenues, not to a block face of Fifth Avenue. In particular, East 86th Street is not returned as a cross street, since it is the true 'on' street; and Fifth Avenue is returned as the cross street at the low-address end of the segment. The Spatial Coordinates returned are as follows. If the vanity street is in reality a cross street at one end of the true street segment, the Spatial Coordinates returned are those of a point calculated under the assumption that the building entrance is the first one occurring on that end of the true block face. In the 1049 FIFTH AVENUE example, the Spatial Coordinates returned are that of a point on the right side of the Fifth Avenue end of the segment of East 86th Street between Fifth and Madison Avenues. If the vanity street is not a cross street at either end of the true segment, or is a cross street at both ends, then the Spatial Coordinates returned are those of a point calculated under the assumption that the building entrance is located at the midpoint of the true block face.

Function 1A processes vanity addresses as follows. A warning with Reason Code 'V' is not issued. The output data returned pertain to the actual building associated with the vanity address. In the list of geographic identifiers at least two entries will appear: one (type V entry) for the vanity address and one for the real street. The second entry will be either a regular address entry with house numbers and street information or a type R entry indicating the street that the entrance to the vanity address is on.

V.10 Out-of-Sequence Addresses

In addition to vanity addresses, there is another type of geographically dislocated address called out-of-sequence addresses. In such an address, the street name does refer to the street where the referenced building entrance is actually located, but the house number is out of sequence with those of the adjacent buildings. An out-of-sequence address may or may not be so dislocated that the building entrance is on a block face other than the one that is consistent with the normal addressing pattern of the given street. Developers sometimes request such addresses because they feel they are euphonious or easy to remember.

An example of an out-of-sequence address is 62 WEST 62 STREET in Manhattan. This address refers to a building entrance located on the south side of West 62nd Street between Broadway and Columbus Avenue. In this case, the out-of-sequence address is indeed on the block face that is consistent with the normal addressing pattern for West 62nd Street. However, the building in question is directly to the east of a building with the address range 42-44 WEST 62 STREET. This violates the normal addressing pattern for West 62nd Street, and for east-west streets on the west side of Manhattan in general, in which the house numbers consistently increase going from east to west.

Functions 1 and 1E process out-of-sequence addresses as follows. A warning with Reason Code ‘O’ is issued for any address on a block face containing an out-of-sequence address. The output data returned, including cross streets and geographic district identifiers, pertain to the block face on which the building entrance is actually located. The Spatial Coordinates returned are those of a point calculated under the assumption that the building entrance is located at the midpoint of the block face.

An opposite-parity address contains a house number that is of the opposite parity to the predominant parity on the block face. Opposite-parity addresses are processed in the same manner as out-of-sequence addresses.

CHAPTER VI: TAX LOT AND BUILDING PROCESSING - FUNCTIONS 1A, BL, BN

VI.1 Introduction

New York City has approximately one million parcels of privately and publicly owned real property, called tax lots, containing more than 800,000 buildings. This chapter describes the Geosupport functions that process tax lots and buildings, Functions 1A, BL and BN.

Two data items discussed in detail in this chapter, the Borough-Block-and-Lot (BBL) and the Building Identification Number (BIN), serve as unique identifiers for tax lots and buildings, respectively. (Addresses are non-unique building identifiers, since many buildings have more than one address.) Function 1A accepts address input, Function BL accepts BBL input, and Function BN accepts BIN input.

VI.2 Tax Lots and BBLs

The city's tax geography is designated and modified by the New York City Department of Finance (DOF). The tax geography consists of the subdivision of the territory of the city (excluding city-owned land that is mapped for streets) into tax blocks, each of which is further subdivided into one or more tax lots.

- Each tax block is identified, uniquely within its borough, by a tax block number assigned by DOF. Each tax block can consist of one, more than one, or a portion of one physical city block.

- Each tax lot is identified, uniquely within its tax block, by a tax lot number assigned by DOF.

Thus, each of the city's tax lots is identified, uniquely within the entire city, by the combination of three items, the borough code, tax block number and tax lot number. These items are often concatenated to form a single data item called the Borough-Block-and-Lot (BBL).

DOF strives to keep the tax block numbering as stable as possible over time, to facilitate property title searches and other historical record-keeping. For example, when a new stretch of street divides what was a single physical block into two physical blocks, DOF generally retains the old tax block number for both of the new physical blocks. As a result, there are many tax blocks that consist of more than one physical block. Occasionally, DOF does subdivide a tax block into two or more new tax blocks, assigning new tax block numbers to them. This may be done when a large area of land is being developed, often in conjunction with the mapping of a new pattern of streets. In recent years, this has most commonly occurred in Staten Island; Battery Park City in Manhattan is another recent instance.

In contrast to the relatively stable tax block geography, the tax lot geography is quite volatile. DOF constantly merges and 'apportions' (subdivides) tax lots, generally assigning new tax lot numbers to the newly created tax lots. However, DOF sometimes reassigns the tax lot number of a 'predecessor' lot (one of the lots that is being merged or apportioned out of existence) to a 'successor' lot. As a result, it is possible for the same BBL value to refer simultaneously to an existing tax lot and to one or more tax lots that no longer exist.

Figure VI-1, below, illustrates the tax geography for a portion of Manhattan in the vicinity of City Hall.

Figure VI-1: Tax Geography for a Portion of Manhattan

The large bold numbers in Figure VI-1 are tax block numbers, and the small numbers are tax lot numbers. Notice that tax block 127 is a case of a tax block consisting of two physical blocks. Also notice that tax blocks 134 and 135 both have a tax lot 9, exemplifying that tax lot numbers are unique only within a tax block. (Similarly, tax block numbers are unique only within a borough.)

<u>VI.3 Buildings and Building Identification Numbers (BINs)</u>

Many city agencies must maintain and process building-related data rather than or in addition to tax lot-related data. These two levels of processing are distinct, since a single tax lot can contain more than one building or no buildings.

A critical issue for building-level processing is to be able to identify buildings <u>consistently</u>. Neither addresses nor BBLs are suitable to serve as consistent identifiers for buildings. Some shortcomings of using addresses as building identifiers are as follows:

- Many buildings have more than one address.
- Some buildings have no addresses.
- The same address can identify both an existing building and a demolished one.
- New York City has a small number of instances in which two different existing buildings have the same address (see Section V.6).

Some shortcomings of using BBLs as building identifiers are as follows:

- Some tax lots contain more than one building.
- The relationship of buildings to tax lots is volatile, since tax lots are often subdivided and merged over time.

In order to provide a unique, immutable, citywide standard for building identification that can support consistent building-level processing, GSS has developed a set of Building Identification Numbers (BINs) that are assigned to every building in the city. (BINs are distinct from, and should not be confused with, house numbers.) A BIN is a seven-byte numeric item, the first digit of which is the borough code. If a BIN field in a Geosupport work area is ‘empty’ (devoid of information), it contains the borough code followed by all zeros (in contrast to most Geosupport fields, which contain all blanks when ‘empty’).

By using BINs as the building identifier, city agencies can process and match building-related data easily and in a consistent manner. Indeed, there are buildings that do not have either an address or a Non-Addressable Place Name (NAP) and can be identified only by their BIN. In this document, such buildings are called Non-Addressable Un-named Buildings (NAUBs). Typical examples of NAUBs are some storage sheds on industrial lots and some comfort stations in parks.

The proliferation of the use of BINs among city agencies facilitates matching data by building across applications and across agencies. The Department of Buildings, which is particularly involved with building-level processing, uses BINs to identify buildings in its major computer applications.

The BIN that is assigned to a building is never changed (except to correct assignment errors); it remains assigned to that building permanently, even if the building is subsequently demolished or its BBL changes as a result of a tax lot merger or apportionment. If a building is demolished, and a new building is subsequently built and given the same address as that of the demolished building, GSS assigns to the new building a new BIN, different from that of the demolished building. In this case, the same address ambiguously identifies two distinct buildings (the new one and the demolished one), but each building is unambiguously identified by its own unique BIN. However, only one BIN per address is ‘active’ in the Geosupport System at one time. That is, only one BIN per address is accepted as Function BN input and returned as Function 1A and BL output. Generally, the active BIN of an address is the BIN assigned to the most recent building at the given address. Function BN also accepts as input the BINs assigned to buildings that have no addresses, such as NAUBs.

VI.4 Condominiums and Billing BBLs

Condominiums are a class of properties with unique characteristics. A condominium consists of condominium units, each of which constitutes a separate tax lot that has its own BBL. In a residential condominium, the condominium units are generally the individual apartments. In a commercial condominium, the units might be retail shops or blocks of space in an office building. There are also mixed-use condominiums that have both commercial and residential units. A condominium can encompass all or part of a building or more than one building, possibly on more than one tax block.

The individual units in a condominium (but not the condominium itself) are parcels of real property. For example, title to an individual unit can be conveyed via a deed; unit owners are responsible for paying real estate taxes directly to the city; and liens can be placed against units.

Many municipal operations relate to condominiums as a whole rather than to specific condominium units.

Examples are collecting sanitation fines, issuing code violations and inspecting and licensing building-wide systems such as boilers and elevators.

To distinguish condominiums from their constituent units, DOF has assigned to each condominium a set of special tax lot identifiers called the billing BBL. (Condominiums are also identified by a Condominium Identification Number, also assigned by DOF.) If a condominium occupies land on more than one tax block, DOF assigns a billing BBL to each portion of the condominium on a separate tax block. Function BL accepts billing BBLs as valid input data, and Functions 1A and BN return them as output data.

DOF assigns billing BBLs only to condominiums, not to condominium units or non-condominium properties. Billing BBLs do not represent actual tax lots, and are not lienable. However, billing BBLs do provide a mechanism to obtain the name and address of a more appropriate party than a unit owner to communicate with concerning condominium-wide matters. DOF maintains files keyed to BBL which, for conventional BBLs, contain the names and addresses of parties registered to receive bills for real property taxes, often the property owner, but possibly a property manager, attorney or mortgagee. For a condominium billing BBL, the party listed might be an officer of the condominium, a property manager or an attorney. (Note: Geosupport does not provide direct access to DOF's files; to arrange such access, contact DOF.)

VI.5 Vacant Street Frontages and Pseudo-Addresses

'Real' addresses are officially assigned to new buildings by the offices of the five borough presidents. In addition, GSS assigns addresses called pseudo-addresses to some vacant street frontages of tax lots, that is, to street frontages that do not have 'real' building addresses. Pseudo-addresses have no 'official' status; they are not meaningful outside of the Geosupport System and should not be used for any operational purpose. In particular, mail sent to a pseudo-address is likely to be undeliverable. Since pseudo-addresses are not associated with buildings, they do not have associated BINs. Note: Pseudo-addresses are not to be confused with the unrelated concept of pseudo-street names (discussed in Section III.6).

When assigning a pseudo-address, GSS attempts to anticipate what 'real' address might someday be assigned to a building if one were to be built at that location. However, the assignment of pseudo-addresses can sometimes involve an element of arbitrariness, especially where there is a wide gap between the two real addresses that 'sandwich' a vacant frontage, or where there is a row of several adjacent vacant frontages. When assigning a pseudo-address, at a minimum, GSS uses a house number that is not already in service on the given street and that is in proper sequence with nearby real house numbers and with any previously assigned pseudo-addresses. If no such house number is available, no pseudo-address is assigned to that vacant street frontage.

Function 1A is designed to accept as input both real addresses and pseudo-addresses. Also, both Function 1A and Function BL include pseudo-addresses in the list of geographic identifiers that they return for a tax lot. Pseudo-addresses serve two purposes:

- Certain information is obtainable from Geosupport by address but not by BBL, such as many political and administrative district identifiers that Functions 1 and 1E return. For vacant tax lots, which have no buildings at all and therefore no real addresses, pseudo-addresses provide the only means to obtain such information from Geosupport. Of course, for those vacant tax lots that have no pseudo-addresses assigned to them, it is not possible to obtain such information from Geosupport.

- By filling in addressing gaps of tax lot frontages along streets, pseudo-addresses enable the possible future development in the Geosupport System of an automated down-the-street 'strolling' function, in which the BBLs of all the tax lots fronting along one side of a street could be retrieved in their proper geographic sequence.

If a pseudo-address comes to be assigned as a real address of a newly constructed building, GSS changes that address's classification in the Geosupport System from pseudo-address to real address. At the same time, the address's tax lot might also change, if the new building happens to be on a different tax lot than the lot to which the address had been assigned as a pseudo-address. Real addresses can also change status and become either Geosupport rejects or pseudo-addresses, as the result of building demolitions. Geosupport is updated to reflect such changes, but time lags are possible.

VI.6 Function 1A

Function 1A processes input addresses and NAPs. When Function 1A is called using two work areas, it returns information in WA2 related to the tax lot and the building (if any) identified by the input data (see work area layouts in Appendix 2 and Appendix 13). If the input address is a pseudo-address, a warning is issued with Reason Code '8' or '9'.

See Chapter V for a general discussion of Geosupport address processing, much of which is applicable to Function 1A. In particular, the various combinations of data items that can be used to specify an input address are described in Section V.3. Section V.4 discusses how Function 1A differs from the other address-processing functions with respect to the validation significance of input address acceptance and rejection, and explains why Function 1A is the best address-processing function to use to validate addresses. Special address processing features discussed in Chapter V are also available with Function 1A, including duplicate address processing, the special Marble Hill/Rikers Island feature, and the special Ruby Street feature.

An important purpose of Function 1A is to provide the BBLs of tax lots for which addresses are known. The BBLs can then be used (outside of the Geosupport System) to retrieve information from various city files that are keyed to BBL, including DOF's billing address files, from which the name and mailing address of the party registered to receive real estate tax bills can be retrieved. This same party might also be an appropriate recipient for many other property-related city mailings, such as notices of inspections, violations, summonses, fines, hearing notifications and licenses.

The long WA2 option is available for Function 1A. Both the regular WA2 and the long WA2 for Function 1A contain a set of data organized into a list. The list in the regular WA2 is a List of Geographic Identifiers (LGI). The LGI is intended to provide a comprehensive geographic profile of the tax lot by listing, so far as the information is known and space allows, all of the buildings the lot contains; all of the street addresses and non-addressable street frontages of each of those buildings; all of the vacant street frontages of the lot; and all NAPs associated with the lot. See the entry for the List of Geographic Identifiers in Appendix 3 for a detailed discussion of the contents of the LGI.

The LGI's entries are ordered so that entries with non-empty BINs are listed first, grouped by BIN. If the input address is a real address, the first group of entries in the LGI consists of the entries for the BIN corresponding to the input address, and (except for certain special cases) the very first entry is an address range encompassing the input address. (The special cases are when the input information contains the alternative borough for a Marble Hill or Rikers Island location or the alternative street name or street code for a Ruby Street address -- see Sections V.7 and V.8.) Any entries with empty BINs, such as entries for pseudo-address ranges, are listed after the entries with non-empty BINs as space in the LGI allows. If the input address is a pseudo-address, an address range encompassing it may or may not appear in the LGI, depending on the availability of space in the LGI and on the order in which the non-BIN entries happen to be listed.

The LGI has a maximum capacity of 21 entries, which for almost all tax lots is sufficient to contain all of the lot's geographic identifiers. A tax lot that does have more than 21 geographic identifiers is said to have the 'LGI overflow condition'. The LGI overflow condition is indicated by a value in the LGI Overflow Flag in Function 1A's regular WA2, as well as by the issuance of a warning (Reason Code 'A').

By definition, when a tax lot has the LGI overflow condition, some of the lot's geographic identifiers are not included in the LGI. In particular, it is possible that the BINs of some of the lot's buildings do not appear in the LGI. The purpose of the long WA2 option for Function 1A is to provide a means for applications to retrieve a complete list of BINs for all the buildings on a tax lot, even for lots that have the LGI overflow condition. The long WA2 contains a List of Buildings on the Tax Lot. Each entry in this list consists only of a BIN; the list includes no address, street frontage or NAP information. (Such information can be obtained for each of the lot's buildings by making Function BN calls.) The maximum capacity of the List of Buildings is 2,500, which is sufficient to list the BINs of all of the buildings on any tax lot in New York City.

Applications can use the long WA2 option for all tax lots, even for those that do not have the LGI overflow condition. Using the long WA2 option for every Function 1A call, rather than only for tax lots that have the LGI overflow condition, may be advantageous in applications that require a list of BINs but do not require the other information returned in the LGI. Doing so simplifies application design: only a single Function 1A call would be required for each input address, and the List of BINs, unlike the LGI, contains every BIN for the lot without repetition. However, users should be aware that, for tax lots that have the LGI overflow condition, a long WA2 call causes Function 1A to perform more I/O operations than a regular WA2 call.

If the input address to a Function 1A call, using either the regular or the long WA2, is that of a condominium, this is indicated by a 'C' in the Condominium Flag. In addition, the following information is returned for condominiums. The BBL field in positions 93-102 for MSW and 102-111 for COW contains the billing BBL of the condominium (or the billing BBL of the portion of the condominium in the tax block containing the input address, if the condominium is in more than one tax block). There is a value in the DOF Condominium Identification Number field, provided that DOF has assigned an ID number to the condominium and GSS has entered it into the Geosupport data. The fields in positions 82-91for MSW and 114-123 for COW and 105-114 for MSW and 125-134 for COW contain, respectively, the low and high BBL of all the condominium units in the building identified by the input address.

<u>VI.7 Function BL</u>

The input to Function BL is a BBL identifying a tax lot. Like Function 1A , Function BL can be called with the long WA2 option.

The layouts of the regular and long WA2s for Function BL are the same as the corresponding layouts for Function 1A, as described in Section VI.5. However, since the input information to Function BL identifies only a tax lot, whereas the input information to Function 1A identifies a specific building, the contents of certain WA2 fields have a different significance for Function BL than for Function 1A. These fields are the BBL (in positions 29-38 for MSW and 34-44 for COW), the BIN (in positions 70-76 for MSW and 82-88 for COW), and the LGI (in positions 184-939 for MSW and 251-1363 for COW) of the regular WA2. All other WA2 fields have the same contents for both functions. For Function BL, the contents of the WA2 fields for the BBL, BIN and LGI are as follows:

- <u>If the input BBL identifies a single-building non-condominium tax lot</u>: The output BBL field contains the input BBL. The BIN field contains the BIN of the tax lot's only building. The LGI may contain all types of entries. As with Function 1A, the LGI may or may not be comprehensive with respect to the tax lot's real address ranges and with respect to its BINs.

- If the input BBL identifies a multi-building non-condominium tax lot: The output BBL field contains the input BBL. The BIN field contains the BIN of an arbitrary one of the tax lot's buildings. The LGI may contain all types of entries. As with Function 1A, the LGI may or may not be comprehensive with respect to the tax lot's real address ranges and with respect to its BINs.

- If the input BBL identifies a vacant tax lot, i.e., a tax lot that has no buildings: The output BBL field contains the input BBL. The BIN field contains the Borough Code followed by all zeros. The LGI consists of all of the pseudo-address ranges (type Q entries) assigned to the tax lot, and all vacant street frontages (type F entries) of the tax lot.

- If the input BBL identifies a condominium unit: The output BBL field contains the billing BBL of the condominium. If the condominium occupies portions of more than one tax block, the output BBL field contains the billing BBL of the portion of the condominium that is specific to the tax block containing the input condominium unit. The BIN field contains the BIN of the building that contains that unit. The LGI contains building-related entries (real address ranges, NAUBs (type B entries), NAPs (type G, N and X entries) and blank-wall building frontages (type W entries)) only for the building containing the input condominium unit, and is comprehensive for that building. The LGI can contain all non-building-related types of entries (vacant street frontages (type F entries) and pseudo-address ranges (type Q entries)).

- If the input BBL is a billing BBL of a condominium: The output BBL field contains the input BBL. The BIN field contains the BIN of an arbitrary one of the condominium's buildings on the tax block identified by the input billing BBL. The LGI may contain all types of entries. As with Function 1A, the LGI may or may not be comprehensive with respect to the tax lot's real address ranges and with respect to its BINs.

VI.8 Standard and Legacy Versions of Functions 1A and BL

The version of Functions 1A and BL that is documented in this User Programming Guide is called the standard version. The standard version was first created in 1995 when major modifications were made to Functions 1A and BL. The predecessor version of Functions 1A and BL is called the legacy version, and is no longer supported with version 10.0. If you have programs that use the legacy PAD, please refer to the Geosupport Technical Bulletin 05-1 dated February 18, 2005 for information on converting to standard PAD. Copies of this bulletin are available upon request to GSS

.

All MSW applications that invoke Functions 1A and BL must set the 1ABL Version Switch to the value 'S'. Note: COW applications only support standard PAD, so the 1ABL Version Switch is not applicable.

VI.9 Function BN

Function BN processes a building specified by an input BIN. Function BN does not have the long WA2 option.

The layout of WA2 for Function BN is identical to that of the regular WA2 for Function 1A. However, in

Function BN’s WA2, the LGI contains entries only for the input building; not for any other buildings on the tax lot. It also does not contain any vacant street frontage (type F) or pseudo-address (type Q) entries. Function BN’s LGI is always complete with respect to the input building, since there is no building in New York City that has more than 21 geographic identifiers.

Except for the difference in the contents of the LGI and its list counter field, the Number of Entries in the LGI, the contents of Function BN’s WA2 and Function 1A’s regular WA2 are identical for a given tax lot.

CHAPTER VII: STREET CONFIGURATION PROCESSING - FUNCTIONS 2, 3, 3C, 3S

VII.1 Introduction

This chapter discusses various types of geographic locations known collectively as ‘street configurations’, and the Geosupport functions that process them. Street configurations are locations that are specified in terms of a combination of either two or three streets.

- The two-street configurations are street intersections, which are specified either in terms of a pair of intersecting streets or in terms of a single intersection name.
- The three-street configurations are locations that are specified in terms of an ‘on’ street between two cross streets. There are three types of three-street configurations: street segments, block faces, and street stretches.

Table VII-1 lists the types of street configurations that Geosupport can process, the data items required to specify each type, the functions that process them, and the sections of this chapter in which they are discussed. The final section in this chapter describes borough boundary processing, a special feature of all the street configuration functions except Function 3S.

Table VII-1: Street Configuration Types and the Functions that Process Them

Street Configuration Type	Specified By	Function	Section
Intersections	2 intersecting streets and, if the 2 streets intersect twice, a compass direction specifying which intersection to process; or, one intersection name	2	VII.2
Street Segments (and related configurations)	‘On’ street and 2 consecutive (or ‘nearly’ consecutive) cross streets	3	VII.4
Block Faces	‘On’ street, 2 consecutive cross streets, and compass direction specifying side of street	3C	VII.5
Street Stretches	‘On’ street and (optionally) any 2 cross streets and, if the ‘on’ street intersects a cross street twice, a compass direction specifying which intersection to process	3S	VII.6

Applications pass input streets to the street configuration functions in the appropriate WA1 input fields, either in the form of street names or in the form of street codes. In the case of two-street configurations, the order of the two input streets is immaterial; either input street may be passed in either WA1 input street field. In the case of three-street configurations, the ‘on’ street must be passed in the WA1 input ‘on’ street field; the two cross streets may be passed in either order in the two WA1 input cross street fields.

For any of the street configuration functions, input street names may be pseudo-street names or intersection names, except for the ‘on’ street in a three-street configuration. Place names may not serve as input street names. (For a discussion of non-street features, pseudo-street names, place names and intersection names, see Section III.6.) <u>In the remainder of this chapter, the term ‘street’ refers to a street name or street code that conforms to the above criteria.</u>

Geosupport processes street configurations based on a simplified model of the city’s geography embodied in a digital map of New York City called LION. LION is a single-line map, that is, it represents streets and other linear geographic features, including railroad lines and shorelines, as single lines with no thickness, and it represents intersections as single points with no area or internal detail. In reality, of course, intersections occupy areas of various sizes and shapes, as reflected in a more realistic type of map known as a double-line map. The distinction between a single-line map and a double-line map is illustrated in Figure VII-1.

Figure VII-1: Single-Line and Double-Line Maps Contrasted

<u>VII.2 Intersections: Function 2</u>

Function 2 is the Geosupport function that processes street intersections. Function 2 can process not only conventional street intersections, but also intersection names and ‘pseudo-intersections’, that is, intersections of a conventional street with a pseudo-street (see Section III.6). There are three types of pseudo-intersections: dead ends, points at which a street intersects with the city limits, and bending points of streets. A point is considered a bending point if the angle of the street at that point is not within the range 160-200 degrees (i.e., not within 20 degrees of a straight line).

<u>Nodes</u>

We use the term <u>node</u> generically to refer to all types of intersections, both conventional and pseudo. In accordance with the LION model of the city’s geography, nodes should be visualized as being single points located approximately at the centers of the intersections they represent. Nodes can serve not only as Function 2 input, but also as the delimiting endpoints of street stretches for input to the functions that process three-street configurations.

Formally, a node is a point along a street where one of the following occurs:

- Conventional intersection of two streets: The street intersects with at least one other street (called a cross street). Example: 'the intersection of BROADWAY and CHAMBERS STREET in Manhattan'

- City Limit Point: The street intersects with the city limits. (The street may terminate at that point or it may continue as a suburban street). City limit points occur only at the Bronx-Westchester County or Queens-Nassau County border. Example (see Figure VII-2): 'LINDEN BOULEVARD at the CITY LIMITS in Queens'

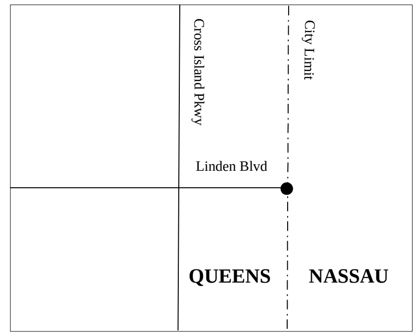

Figure VII-2: City Limit Point

- Dead End: The street has a termination point (called a dead end) that is not at the city limits and at which there are no cross streets. Example (see Figure VII-3): 'DEAD END of CROES AVENUE in the Bronx'. (City limit points are excluded from being treated as possible dead ends because city streets may continue across the city limits into the adjacent suburban county.)

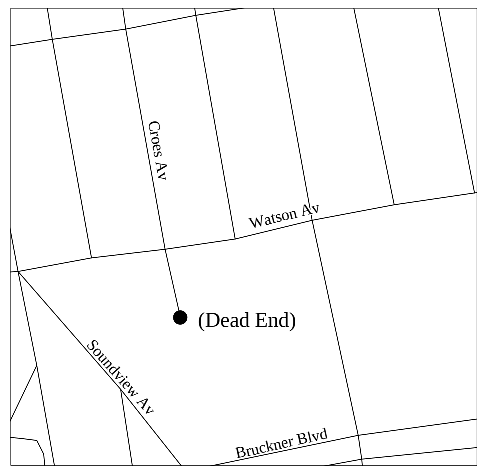

Figure VII-3: Dead End

- Bend: The street has a bending point. Examples (see Figure VII-4): 'BEND of COMMERCE STREET' and 'BEND of BARROW STREET' in Manhattan. Note that the bending point of Barrow Street is also a conventional street intersection, the intersection of Barrow and Commerce Streets. The Commerce Street bending point is not a conventional intersection, and can only be specified in terms of the pseudo-street BEND.

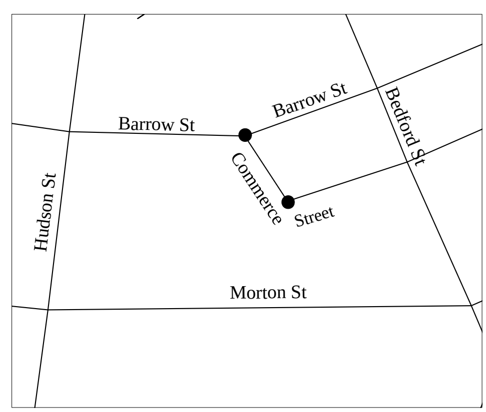

Figure VII-4: Bends

Conceptually, nodes can be characterized as those points along streets that can be specified in a form recognizable to Geosupport, that is, in the form of an intersection name, or a pair of street codes, or a pair of street names that possess street codes. One and only one street of the pair may be a pseudo-street. The allowable pseudo-streets are DEAD END and its aliases, CITY LIMITS and its aliases, and BEND and its aliases. See Section III.6 for a discussion of pseudo-street names.

Number of Intersections of a Pair of Streets

Given any pair of New York City streets (or a street and a pseudo-street), there are four possibilities:

- The two streets do not intersect at all.
- They intersect at one location (the 'unique-node case').
- They intersect twice (the 'two-node case').
- They intersect more than twice (the 'many-node case').

Function 2 can process a pair of input streets in the unique-node case and in the two-node case, but not in the many-node case. However, aside from dead ends and bends, the many-node case is very rare. There are numerous streets that have more than two dead ends, and there are numerous streets that have more than two bends.

The Two-Node Case

The two-node case occurs with greater frequency than might be expected. Some types of situations in which the two-node case occurs are the following:

- A street bends or curves, causing it to intersect with a second street at two different points. An example in Queens is the two intersections of the curved street Cromwell Crescent with Alderton Street (see Figure VII-5).

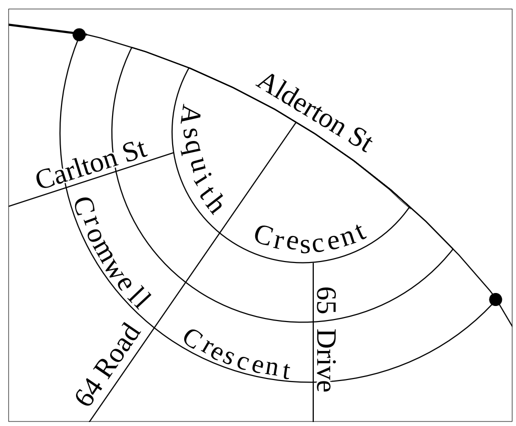

Figure VII-5: Street Intersecting Twice with Curved Street

- A street has a displacement or offset as it crosses another street (a configuration sometimes called a 'dogleg'), so that there are two points where the two streets intersect. An example in Brooklyn is Ditmas Avenue where it crosses Coney Island Avenue (Figure VII-6).

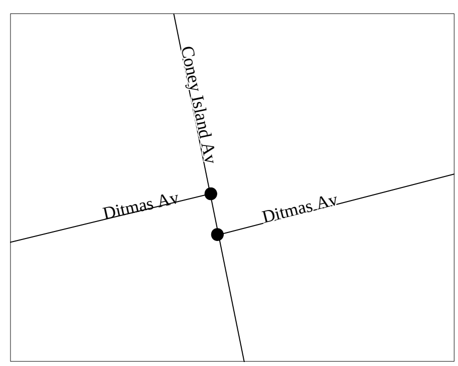

Figure VII-6: 'Dogleg'

- A street forks into two branches (for example, around a traffic island, plaza or small park) such that both branches have the same street name and they both intersect with another street. An example in Manhattan is Duane Street, which forks around a small triangular park; both branches of Duane Street intersect with Hudson Street (Figure VII-7).

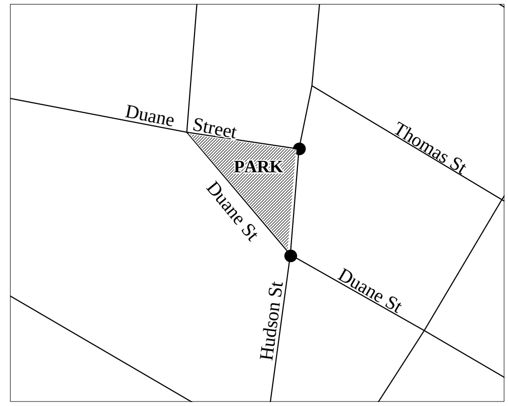

Figure VII-7: Street Fork

- A street has exactly two dead ends. An example in the Bronx is Odell Street (Figure VII-8). The two pseudo-intersections of Odell Street with the pseudo-street DEAD END are considered to be an instance of the two-node case. Similarly, streets that have exactly two intersections with the city limit, and streets that have exactly two bending points, are instances of the two-node case.

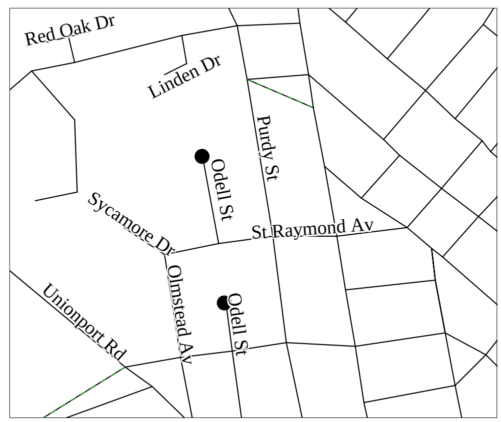

Figure VII-8: Street with Two Dead Ends

<u>Specifying Function 2 Input Data</u>

Applications pass an input node to Function 2 by specifying either an intersection name or two distinct streets (i.e., two streets that have different B5SCs) identifying the intersection. If the input information is in the form of an intersection name, it may be passed in either WA1 input street field, and the other field should preferably be left blank or it may contain any conventional street that exists at the given intersection. If the input data are in the form of two streets that are an instance of the two-node case, an input compass direction ('N', 'S', 'E' or 'W') must also be specified. The compass direction identifies which of the two nodes is to be processed, by specifying that node's spatial position relative to the other one. For example, if 'N' is specified as the input compass direction, then Function 2 will process the northernmost of the two nodes.

Note that an intersection may be an instance of the two-node case when specified (with a compass direction) in terms of a particular pair of streets, while the same intersection may be an instance of the unique-node case when specified (without a compass direction) in terms of a different pair of streets.

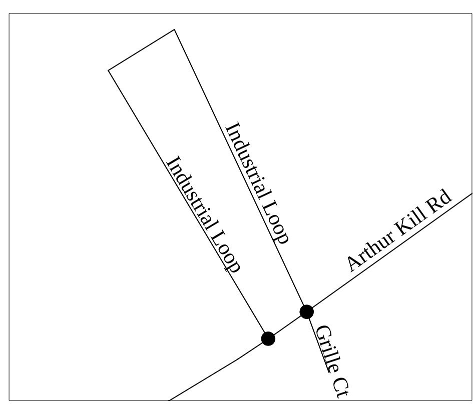

Figure VII-9: Simultaneous 2-Node and Unique-Node Case

For example, in Staten Island, Industrial Loop and Arthur Kill Road intersect at two nodes (see Figure VII-9). When specifying either of those nodes as an intersection of Industrial Loop and Arthur Kill Road, a compass direction is required. However, at the northern (or alternatively, the eastern) one of those nodes, there is a third street, Grille Court. That node can be specified, without a compass direction, as the unique intersection either of Grille Court and Industrial Loop, or of Grille Court and Arthur Kill Road.

In some instances of the two-node case, the two nodes are 'nearly' (i.e., within 10 compass degrees of) due east-west of each other. In that event, attempting to distinguish between the two nodes in terms of the compass directions north and south would be highly error-prone. In those cases, therefore, Function 2 accepts only 'E' and 'W' as valid input compass directions. Similarly, if the spatial relationship between the two nodes is 'nearly' (within 10 degrees of) due north-south, Function 2 accepts only 'N' and 'S'.

For Manhattan only, in the two-node case, Geosupport rotates the spatial relationship between the pairs of nodes 30 degrees counterclockwise before determining whether they are 'nearly' north-south or east-west of each other. This comports with the widespread conventional treatment of the avenues and streets in most of Manhattan as if they were oriented due north-south and due east-west, respectively. In reality, Manhattan's principal street pattern lies approximately at a 30-degree clockwise rotation from the cardinal points of the compass. For a more detailed discussion of the 30-degree rotation for Manhattan, see the entry for Segment Orientation in Appendix 3.

In most instances of the two-node case, the two nodes have a pronounced 'diagonal' spatial relationship, that is, they are not within 10 degrees of either due north-south or due east-west of each other. In that case, Function 2 accepts all four compass directions as valid input. For example, either 'N' or 'W' is accepted as a specification for the northwestern intersection of Alderton Street and Cromwell Crescent (the intersection highlighted on the upper left in Figure VII-5); either 'S' or 'E' is accepted as a specification for the southeastern (lower right) intersection of these streets.

Possible Outcomes of a Function 2 Call

Table VII-2 lists possible outcomes of a Function 2 call by Geosupport Return Code (GRC). Table VII-2 does not include standard reject conditions that are applicable to most Geosupport functions, such as an inability to normalize or recognize an input street name. In Table VII-2, the term 'intersection' also encompasses pseudo-intersections.

Table VII-2: Possible Outcomes of a Function 2 Call

GRC/Reason Code	Meaning
00	(Successful call) If the input information was in the form of an intersection name, it was recognized as a valid name of a specific intersection. If the input information was in the form of two streets, they intersect once or twice, and if twice, an input compass direction has been supplied which is a valid descriptor for one of those intersections. A full complement of output data is returned in the work areas.
01/H	(Warning) The two input streets intersect once, but the input compass direction field is non-blank. The input compass direction field is ignored. A full complement of output data is returned in the work areas.
01/T	(Warning) The input street name is ignored if an intersection name is specified along with a street name that is part of the intersection.
02	(Reject) The two input streets intersect twice, but the input compass direction field is blank. A valid input compass direction value is required for these input streets.
03/3 thru 9	(Reject) The two input streets intersect more than twice. Geosupport cannot process such intersections. The Reason Code value is the number of intersections of the two streets; the value '9' signifies nine or more intersections.
12	(Reject) The input information was in the form of an intersection name or a street code of an intersection name. Geosupport recognizes this name or code as valid, but does not yet have this name or code associated with a specific intersection.
30	(Reject) An input intersection name was specified along with an input street name, but the input street is not part of the intersection.
39	(Reject) The input compass direction field contains a non-blank value other than 'E', 'N', 'S' or 'W'.
40	(Reject) The two input streets intersect twice, but the two intersections are nearly N-S or E-W of each other and the input compass direction is an invalid descriptor for either of the intersections.
62	(Reject) If the input data were in the form of two input street names or codes, the two input streets do not intersect.

Function 2 Output Data

Among Function 2's WA2 output items are the following:

- Identifiers for a set of geographic districts that contain the intersection, including Census Tract, Community District and Police Precinct.

- Spatial Coordinates of the intersection. These correspond to a nominal center point of the intersection and should not be treated as a precision identification of any particular point location on the earth's surface. (For a more detailed discussion, see the Spatial Coordinates entry in Appendix 3.)

- A List of Intersecting Streets (in the form of PB5SCs for MSW and B5SCs for COW) identifying up to five streets incident upon the intersection. Subject to the space limitation, the list may include the PB5SCs or B5SCs of the two input streets, unless one is the pseudo-street BEND, which for Function 2 is never included in the list. The list may also include the PB5SCs or B5SCs of the pseudo-streets CITY LIMIT and DEAD END, and it may include the PB5SCs or B5SCs of any intersection names that are valid for the given intersection. If the application has a need to display the street names of the intersecting streets, the Cross Street Names Flag in WA1 can be turned 'on' and the names will be returned in the List of Street Names in WA1 (see entries for Cross Street Names Flag and List of Street Names in Appendix 3). Note that the cross street names feature incurs processing overhead, and should only be used when necessary.

- A Compass Direction for Intersection Key. If the first two entries in the List of Intersecting Streets are an instance of the two-node case (i.e., they intersect twice), the Compass Direction for Intersection Key contains a compass direction value identifying the intersection in terms of those two streets. If the two streets are not an instance of the two-node case, this field is blank. If both a 'longitudinal' compass direction ('N' or 'S') and a 'latitudinal' compass direction ('E' or 'W') are valid for this intersection, the longitudinal compass direction value appears in this field.

Since Function 2 treats street intersections as if they were single points, Geosupport does not provide a means for a user to request information specific to a portion of an intersection, such as a particular block corner. In the event that an intersection lies on a boundary of two or more geographic districts of a particular type, Function 2 returns the identifier for one of those districts, but provides no indication that some of the intersection's corners may lie in other districts. The district identifier that is returned for such an intersection is selected arbitrarily, but is the same no matter how the intersection is specified.

For example, the intersection of East 116 Street and Fifth Avenue in Manhattan lies on the boundaries of three different Community School Districts (CSDs) (see Figure VII-10). Two of the four block corners at this intersection lie in CSD 3, one lies in CSD 4 and one lies in CSD 5. For this intersection, Function 2 returns CSD 5, and provides no indication that the intersection lies on a CSD boundary.

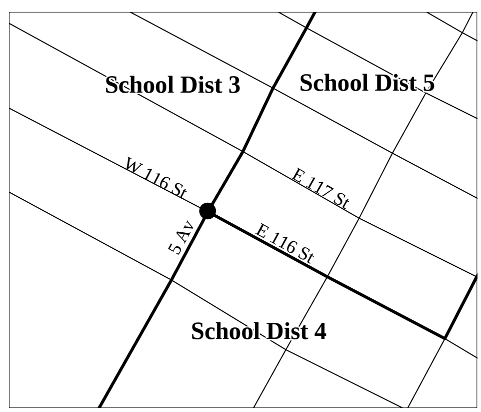

Figure VII-10: Multiple Districts at an Intersection

Multi-Street Intersections and Retrieval Consistency

This subsection discusses the common situation of intersections at which there are more than two streets. Geosupport accepts any pair of those streets as a valid input specification for the intersection. (The concept of an intersection of multiple streets should not be confused with the concept of two streets that intersect at multiple points, which was discussed in the preceding subsection.)

For example, consider the three-street intersection of Hudson Street, Chambers Street and West Broadway in Manhattan (Figure VII-11). The user can specify this intersection to Function 2 in three ways: as the intersection of Hudson Street and Chambers Street; Hudson Street and West Broadway; or Chambers Street and West Broadway.

Similarly, a four-street intersection can be specified in six ways, etc. Function 2 returns identical WA2 information (other than that related to which streets were the input streets for the call) for an intersection regardless of which pair of streets are used to specify it.

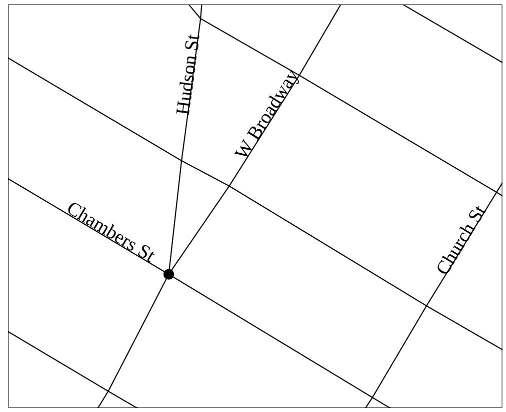

Figure VII-11: Three-Street Intersection

Since an intersection of more than two streets can be specified in more than one way, an important issue for some applications that process data by intersection is the ability to retrieve or match data from an application file consistently by intersection. (For a general discussion of the concept of application file geographic retrieval consistency, see Section I.3.) That is, it is desirable for applications to be able to retrieve data successfully for a multi-street intersection regardless of which pair of streets is used to identify the intersection. A data item called the LION Node Number, which is returned in Function 2's WA2, is designed to serve effectively as a unique, consistent intersection identifier. It is able to so serve because the same LION Node Number is returned regardless of how the intersection is specified. Moreover, the LION Node Number assigned to an intersection is kept constant over time. That is, the same LION Node Number is returned for a given intersection by every Geosupport release, even in cases where there is a change in the set of streets defining an intersection (such as the presence of a new street, the closure of an existing street, or a change in the street code assigned to a street).

Note Concerning the 'Vestigial' Function 2C: A Geosupport enhancement that was implemented in Version 9.5 (March 1998) enables Function 2 to process pairs of streets that intersect twice, using the input compass direction field to identify the specific intersection to be processed. Prior to that enhancement, Function 2 could only process pairs of streets that intersect once, and a separate function, Function 2C, had to be used to process pairs of streets that intersect twice. The enhancement enables Function 2 to process both types of intersection input, rendering Function 2C obsolete. Function 2C is a 'vestigial' function, in the sense in which this term is described in Section I.5. In particular, all new applications should be designed to perform all intersection processing using Function 2 only. It is recommended that users modify existing applications by replacing all Function 2C calls with Function 2 calls. To do so, it may be necessary or appropriate to modify the application's reject handling routines to reflect the situations and GRC's delineated in Table VII-2. Function 2C is not further documented in this User Programming Guide.

VII.3 Three-Street Configurations - Concepts and Terminology

There are many applications in which geographic locations to be processed are identified in terms of an 'on' street between two cross streets. Geosupport can process several types of such locations, namely street segments, block faces and street stretches. We refer generically to all these types of locations as three-street configurations.

This section introduces concepts and terminology needed to discuss three-street configurations. The three succeeding sections discuss the Geosupport functions that process the various types of three-street configurations:

- Section VII.4 discusses Function 3, which processes street segments
- Section VII.5 discusses Function 3C, which processes block faces
- Section VII.6 discusses Function 3S, which processes street stretches.

The definitions below are based on Geosupport's single-line map model of the city's geography, as explained in Section VII.1. Also, recall that the term 'street' refers to a street name or street code that satisfies the criteria delineated in Section VII.1. The term 'node' is as defined in Section VII-2.

Street Stretches and Street Segments

A street stretch is a portion (possibly all) of a street (called the 'on' street) between any two nodes along it (called the delimiting nodes of the stretch). A street stretch is considered to comprise both sides of the 'on' street.

A street segment is a street stretch such that the two delimiting nodes are consecutive along the 'on' street. Every segment is uniquely identified by a segment id. Every street stretch is composed of a set of one or more street segments, which do not necessarily form a continuously connected chain. That is, a street stretch can encompass gaps in the street.

Some examples of street stretches and segments follow.

- The stretch of Madison Avenue between East 51st Street and East 52nd Street in Manhattan (see Figure VII-12) is a street segment. The stretch of Madison Avenue between East 51st Street and East 54th Street is not a segment because its delimiting nodes are not consecutive along the 'on' street; it is a stretch consisting of three segments.

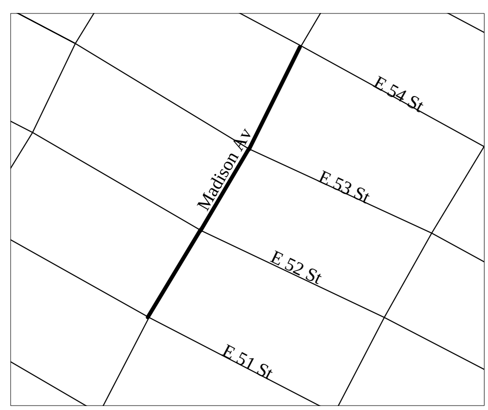

Figure VII-12: Street Stretch

- An example of a street stretch that is not connected is Manhattan's West 64th Street between Central Park West and West End Avenue (see Figure VII-13). West 64th Street has a gap (does not exist) between Columbus Avenue and Amsterdam Avenue, where it is interrupted by Lincoln Center. As a result, the stretch in question consists of two sub-stretches that are not connected to each other.

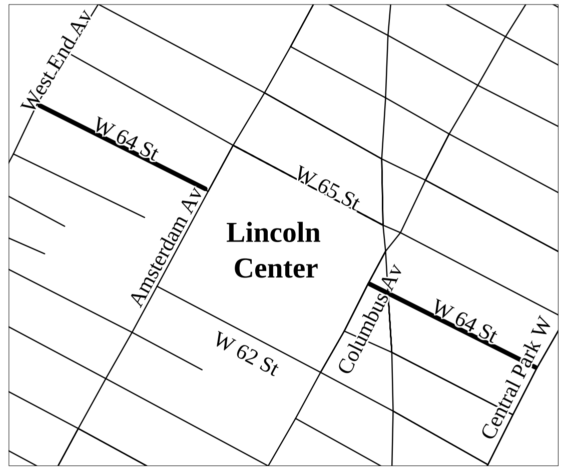

Figure VII-13: Street Stretch Containing Gap

Street stretches can be delimited by pseudo-intersections and intersection names as well as conventional street intersections:

- An example of a street segment in the Bronx delimited by a dead end is CROES AVENUE between WATSON AVENUE and DEAD END (see Figure VII-3).

- An example of a street stretch in Queens delimited by the city limits is LINDEN BOULEVARD between CROSS ISLAND PARKWAY and CITY LIMITS (see Figure VII-2).

- Some examples of street segments in Manhattan delimited by bends are: BARROW STREET between HUDSON STREET and BEND (also specifiable in terms of conventional streets as BARROW STREET between HUDSON STREET and COMMERCE STREET); and COMMERCE STREET between BARROW STREET and BEND (the only way to specify this segment) (see Figure VII-4).

Logical Direction Assigned to Streets

GSS has assigned a logical direction to every street segment in New York City. References to the left and right sides of any segment, and references to its delimiting nodes as the 'from' node and 'to' node, are relative to the segment's logical direction.

For streets that have addresses, the logical direction is always assigned as the direction of increasing addresses. Therefore, for any street with addresses, the 'from' node of any segment is always the node at its low address end, and the 'to' node is the node at the high address end; the left and right sides of the segment are determined accordingly.

For features that have no addresses, such as all railroad tracks and some highways, the logical direction is assigned arbitrarily, but consistently, along the feature's full extent. Note that a street's logical direction, and thus the meaning of 'from', 'to', 'left' and 'right', is unrelated to the street's traffic direction, to its orientation with respect to the points of the compass, or to the order in which cross streets delimiting a stretch are specified.

Block Faces

A block face is a continuous frontage of a physical city block along one street, ignoring the presence of any bending points. That is, the portions of a street frontage of a block that lie on both sides of a bending point are considered to be parts of the same block face.

For example, the Manhattan block bounded by Madison and Park Avenues and East 51st and East 52nd Streets has the following four block faces (see Figure VII-14, which, unlike most of the figures in this chapter, contains a double-line map to illustrate clearly the concept of a block face):

- The east side of Madison Avenue between East 51st and East 52nd Streets
- The south side of East 52nd Street between Madison and Park Avenues
- The west side of Park Avenue between East 51st and East 52nd Streets
- The north side of East 51st Street between Madison and Park Avenues

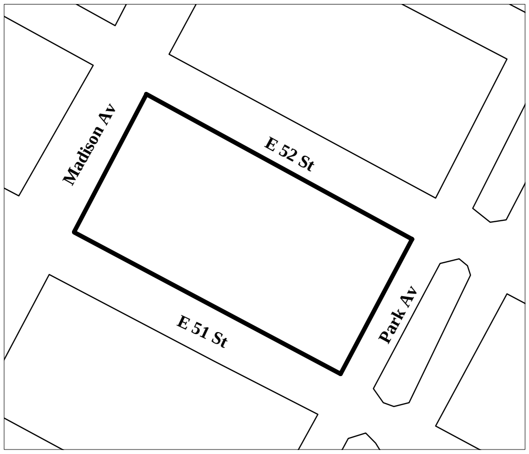

Figure VII-14: Block with Four Block Faces

An example of a stretch with a bending point in Manhattan is Commerce Street between Bedford and Barrow Streets (see Figure VII-4). Both sides of this stretch are single block faces, even though the stretch consists of two segments connected at the bending point.

T-Intersections

A street stretch, and in particular a street segment, is considered to comprise both sides of the 'on' street. In the case of a street segment, each side necessarily is either a single entire block face or a portion of one. Many segments consist of a pair of facing entire block faces along the 'on' street. However, this is not the case at a street configuration called a T-intersection. A T-intersection (so named because the streets are configured like the letter 'T') is an intersection where a cross street intersects the 'on' street on one side of the 'on' street only, and there are no cross streets on the other side of the 'on' street at that intersection. At a T-intersection, the 'on' street has a block face that encompasses more than one segment, and conversely, at least one side of each of those segments consists of only a portion of that block face.

An example of a T-intersection in Manhattan is the intersection of Fifth Avenue and East 41st Street (Figure VII-15). Because of the presence of the New York Public Library main building along the west side of Fifth Avenue between West 40th and West 42nd Streets, there are no cross streets on the west side of Fifth Avenue where it intersects with East 41st Street. The long block face on the west side of Fifth Avenue encompasses two segments, each consisting of a portion of this long block face facing a shorter entire block face on the east side of Fifth Avenue.

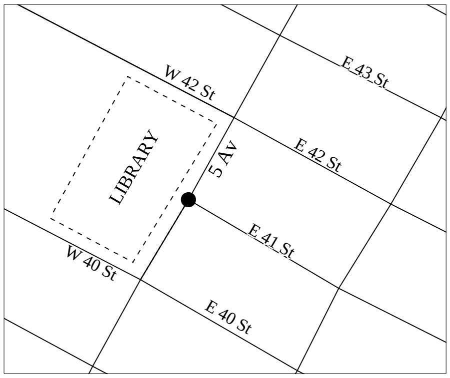

Figure VII-15: T-Intersection

It is possible for a street to have T-intersections at several consecutive nodes. An example in Manhattan is the three-segment stretch of Lexington Avenue between East 42nd and East 45th Streets (Figure VII-16). Because of the presence of the Grand Central Terminal complex on the west side of Lexington Avenue, East 43rd Street and East 44th Street intersect Lexington Avenue only on its east side, forming two consecutive T-intersections along Lexington Avenue. As a result, the west side of this stretch is a single long block face, which faces three shorter block faces on the east side of Lexington Avenue.

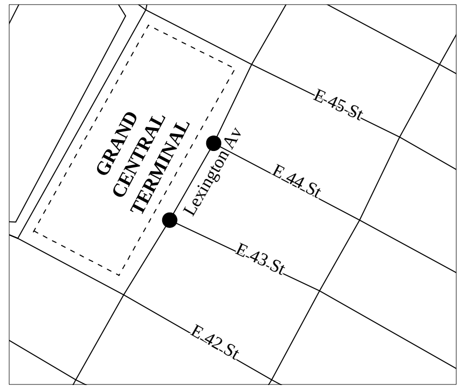

Figure VII-16: Two Consecutive T-Intersections

When a street has consecutive T-intersections at which the cross streets are on alternating sides of the 'on' street, then long block faces on both sides of the 'on' street face each other in overlapping fashion, forming segments both sides of which consist of portions of those long block faces. Union Avenue in Staten Island is an example (Figure VII-17). Note that Union Avenue between Leyden Avenue and Walloon Street is a street segment, because the two delimiting intersections are consecutive along Union Avenue, even though the cross streets are on opposite sides of the 'on' street.

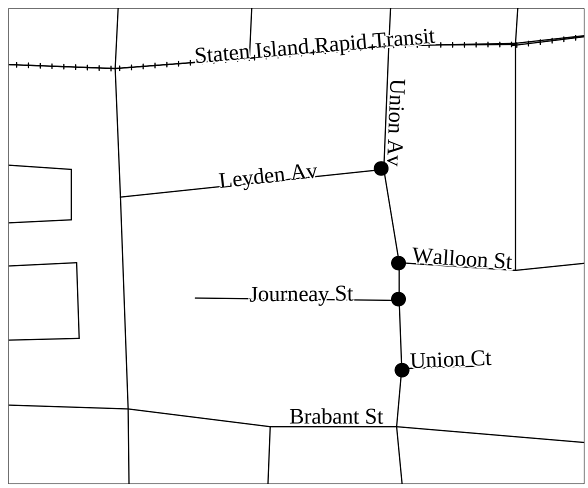

Figure VII-17: T-Intersections on Alternating Sides of Street

<u>VII.4 Street Segments and Related Configurations: Function 3</u>

Function 3 is designed to accept as input portions of a street that are, loosely speaking, 'one block long'. More precisely, Function 3 processes two types of input street stretches:

- Street segments (i.e., portions of a street between two consecutive nodes).
- Street stretches consisting of more than one segment, such that at least one side of the street stretch is a single entire block face. This case has two sub-cases:
  - At a T-intersection, there is a street stretch in which one side is a single entire block face.
  - At a bending point at which there are no cross streets, there is a street stretch in which both sides are single entire block faces.

The following examples illustrate the types of input data acceptable and not acceptable to Function 3. For the Lexington Avenue (in Manhattan) examples, see Figure VII-16. For the Union Avenue (in Staten Island) examples, see Figure VII-17. For the Commerce Street (in Manhattan) examples, see Figure VII-4. For the Croes Avenue (in the Bronx) example, see Figure VII-3.

Input 'On' Street	**One Cross Str.**	**Other Cross Str.**	**Fn 3 Action**	**Reason for Action**
Lexington Avenue	East 42 Street	East 43 Street	Accepted	Single segment
Lexington Avenue	East 43 Street	East 44 Street	Accepted	Single segment
Lexington Avenue	East 44 Street	East 45 Street	Accepted	Single segment
Lexington Avenue	East 42 Street	East 45 Street	Accepted	Single entire block face on west side of Lexington Ave
Lexington Avenue	East 42 Street	East 44 Street	Rejected	Multi-segment, neither side of Lex. Ave is a single entire block face - west side is a portion of a block face, east side comprises 2 block faces.
Lexington Avenue	East 43 Street	East 45 Street	Rejected	Multi-segment, neither side of Lex. Ave is a single entire block face - west side is a portion of a block face, east side comprises 2 block faces.
Union Avenue	Leyden Avenue	Walloon Street	Accepted	Single segment
Union Avenue	Walloon Street	Journey Street	Accepted	Single segment
Union Avenue	Journey Street	Union Court	Accepted	Single segment
Union Avenue	Union Court	Brabant Street	Accepted	Single segment
Union Avenue	Leyden Avenue	Journey Street	Accepted	Single entire block face on west side of Union Avenue
Union Avenue	Walloon Street	Union Court	Accepted	Single entire block face on east side of Union Avenue
Union Avenue	Journey Street	Brabant Street	Accepted	Single entire block face on west side of Union Avenue
Commerce Street	Barrow Street	Bend	Accepted	Single segment
Commerce Street	Bedford Street	Bend	Accepted	Single segment
Commerce Street	Barrow Street	Bedford Street	Accepted	Both sides are single entire block faces
Croes Avenue	Watson Avenue	Dead End	Accepted	Single entire segment

Function 3 Input Data Specification and Validation

Applications pass an input stretch to Function 3 by specifying three input streets, consisting of the 'on' street and two cross streets, in the appropriate WA1 input fields. The input cross streets, but not the 'on' street, may be pseudo-streets or intersection names. The input cross streets may be specified in either order. As with all Geosupport street input, the three input streets to a Function 3 call are specified in the form of either street names or street codes.

If either or both of the delimiting intersections of the input stretch has more than one cross street, the stretch may be specified using any of those cross streets. For example, the segment of Chambers Street illustrated in Figure VII-11 may be specified either as 'Chambers Street between Church Street and West Broadway', or as 'Chambers Street between Church Street and Hudson Street'.

A successful two-work-area call to Function 3 signifies that the three input streets form a combination of an 'on' street and two cross streets that specify either a valid street segment or a valid street stretch at least one side of which is a single entire block face.

Ambiguous Function 3 Input Data

Some combinations of an 'on' street and two cross streets are ambiguous as Function 3 input data, that is, the data specify more than one stretch that satisfies Function 3's input criteria. An example of ambiguous Function 3 input data in Queens is 'Alderton Street between Asquith Crescent and 64th Road' (Figure VII-18). This combination of streets describes two different segments of Alderton Street.

Function 3 rejects such ambiguous input. Unlike Function 2, which provides a means (compass direction input) for users to specify unambiguously an intersection of two streets that intersect in two different places, Geosupport provides the user with no recourse when Function 3 rejects an input stretch specification as ambiguous.

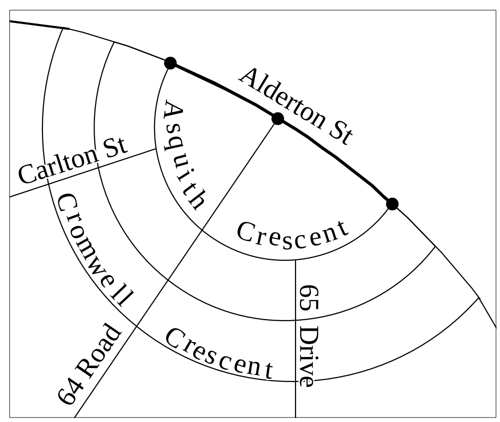

Figure VII-18: Ambiguous Segment Specification

Function 3 Output Data

If a two-work-area call to Function 3 is successful, information about both sides of the input stretch is returned in WA2. (Note that the long WA2 option is available for Function 3. See Section II.5.) Many of the data items in Function 3's WA2 (both regular and long) are paired, with one item for the left side of the 'on' street and another item of the same type for the right side. For example, there are fields for left and right zip code, for left and right 1990 census tract, and for left and right address ranges (each range consisting of a 'from' house number and a 'to' house number). As explained in Section VII.3, left and right are determined by the 'on' street's logical direction, and therefore are independent of the order in which the user specifies the input cross streets.

In the case of an input stretch encompassing more than one segment (the T-intersection and bend cases), the values of the WA2 items that Function 3 returns for the side of the street comprising more than one block face are as follows. The low and high house number values that are returned correspond to the entire stretch. The values that are returned for all other side-related items correspond to the 'last' (relative to the stretch's logical direction) block face. For example, consider Fifth Avenue in Manhattan between East 40 and East 42

Streets (see Figure VII-15). Since the direction of increasing addresses along Fifth Avenue is from south to north, that is also Fifth Avenue's logical direction. It follows that the right side of the given stretch is the east side. It consists of two block faces. Relative to the logical direction, the 'last' of these block faces is the one between East 41 and East 42 Streets. Accordingly, the right address range that is returned in WA2, consisting of the right low house number and the right high house number, corresponds to the entire right side of Fifth Avenue between East 40 and East 42 Streets. The values returned for all other items for the right side of the input stretch correspond to the 'last' block face, the one between East 41 and East 42 Streets. The data returned represents two segments, but only one segment id is capable of being returned; consequently the segment id with the lowest value is returned. If the user request Fifth Avenue between East 40th street and East 41st Street, the segment id returned will be 00034174. If the user requests Fifth Avenue between East 41th Street and East 42nd Street, the segment id returned will be 00034176. However if the user requests Fifth Avenue between East 40th Street and East 42nd Street, which represents two segments, the segment id returned will be 00034174, which is numerically the lower of the two numbers.

Function 3 returns the Segment Length in WA2. This value is expressed in feet, and is computed from the Spatial Coordinates of the LION nodes that constitute the segment's endpoints; it is an approximation to the true length of the segment. <u>Segment Length values provided by Geosupport should not be used in applications that require an engineering level of precision</u>. In the case of an input stretch encompassing more than one segment, the Segment Length value that function 3 returns is the sum of the lengths of the constituent segments.

Another item that Function 3 returns in WA2 is called the Curve Flag. It indicates whether the input segment is curved, and if so, whether that curve is an arc of a circle or is an irregular curve. In the case of an arc of a circle, the Curve Flag indicates on which side of the segment's 'secant line' (the straight line joining the segment's endpoint nodes) the curve lies. If the input segment is curved, whether regularly or irregularly, the Segment Length value returned is approximately equal to the true arc length of the curve, rather than the secant length. If the input stretch encompasses more than one segment, the Curve Flag is returned with an 'on' (non-blank) value if at least one of the constituent segments is curved. For further information, see the entries for Curve Flag and Segment Length in Appendix 3.

The long WA2 option is available for Function 3. The additional data provided in the long WA2 include the census geography and the administrative fire district geography.

<u>Cross Street Reversal Flag</u>

A WA2 item returned by Function 3 called the Cross Street Reversal Flag indicates whether the order of the input streets is consistent with, or opposite to, the stretch's logical direction. This flag can be used to determine which side of the street is the left side and which side is the right side in relation to the order of the input cross streets, as follows:

- If the Cross Street Reversal Flag is returned as a blank, the cross street that was specified in the input First Cross Street field in WA1 is at the 'from' end of the stretch and the cross street that was specified in the Second Cross Street field is at the 'to' end, so that left and right are consistent with facing from the first cross street to the second cross street.

- If the Cross Street Reversal Flag contains an 'R', the first input cross street is at the 'to' end and the second is at the 'from' end, so that left and right are consistent with facing from the second cross street to the first cross street.

For example, consider the segment of Lexington Avenue between East 42nd and East 43rd Streets. Since the addresses on Lexington Avenue increase from south to north, East 42nd Street is at the ‘from’ end of this segment and East 43rd Street is at the ‘to’ end. If an application specifies this segment to Function 3 by passing East 42nd Street in the First Cross Street WA1 field and East 43rd Street in the Second Cross Street WA1 field, the Cross Street Reversal Flag will be returned as a blank. On the other hand, if East 43rd Street is passed in the First Cross Street field and East 42nd Street is passed in the Second Cross Street field, the flag will be returned containing an ‘R’.

Segment Orientation

In applications that involve field operations, such as those that generate work orders to dispatch personnel to specific block faces, the use of ‘left’ and ‘right’ as descriptors of the sides of a street can cause confusion for personnel in the field, because their significance is based on the street’s logical direction, which is not necessarily obvious in the field. Compass directions, on the other hand, are absolute descriptors of the sides of a street; the west side of Lexington Avenue in Manhattan is a specific, invariant side of that street, and is independent of how an observer may be facing. So in such applications, compass directions may be more suitable as side-of-street descriptors than ‘left’ and ‘right’.

To determine compass direction descriptors for the left and right sides of a segment, applications can use an item called the Segment Orientation that Function 3 returns in WA2. The Segment Orientation indicates how the input segment (or the last segment of the input stretch, if it consists of more than one segment) is oriented with respect to the points of the compass, taking into consideration the ‘on’ street’s logical direction. For example, if the input segment is Lexington Avenue between East 42nd Street and East 43rd Street, using the Segment Orientation, the application can determine that the left side of this segment (as determined by the street’s assigned logical direction) is the west side and the right side is the east side. See the entry for Segment Orientation in Appendix 3 for details on the possible values of the Segment Orientation and how to use it to determine compass direction descriptors for sides of streets.

In addition, in some applications it is desirable to be able to describe the ‘ends’ of a street segment in terms of a compass direction.

The application can use the Segment Orientation in conjunction with the Cross Street Reversal Flag to express a compass direction descriptor for a side of a street in terms of left or right relative to a particular ordering of the cross streets (as opposed to left and right relative to the street’s logical direction). For example, the application can determine that, when facing from East 42nd Street to East 43rd Street, the left side of Lexington Avenue is the west side; or that, when facing from East 43rd Street to East 42nd Street, the left side of Lexington Avenue is the east side.

Applications can use the Segment Orientation and the Cross Street Reversal Flag to determine compass direction descriptors for the ‘ends’ of a street segment, as well as its sides. For example, it can be determined that East 43rd Street is at the north end of the Lexington Avenue segment in our example, and East 42nd Street is at the south end.

See the entry for Segment Orientation in Appendix 3 for further details.

Consistent Retrieval of Application Data by On Street and Two Cross Streets

If a street stretch has a delimiting node at which there is more than one cross street, the stretch can be specified in more than one way. Some applications require the ability to retrieve records for stretches from an application file consistently by geographic location, that is, independently of which cross streets were used to specify a stretch at record creation time and which cross streets are used at retrieval time. Function 3 provides items that can be used to form a key for such retrieval. Among the output items that Function 3 returns in WA2 are two lists of street codes (in the form of PB5SCs) for all the cross streets at the two delimiting nodes. Each list is ordered so that its first entry is always the numerically smallest PB5SC of all the entries in that list. This arrangement facilitates the formation by the application of a key for consistent geographic retrieval. The key would consist of a combination of the PB5SC for the 'on' street, the first entry in the 'from' cross street list, and the first entry in the 'to' cross street list.

If the application has a need to display the street names of the cross streets, the Cross Street Names Flag in WA1 can be turned 'on' and the names will be returned in the List of Street Names in WA1 (see entries for Cross Street Names Flag and List of Street Names in Appendix 3). Note that the cross street names feature incurs processing overhead, and should only be used when necessary.

Converting Address-Keyed Application Data to On Street/Cross Streets

As mentioned in Section V.5, Function 1's WA2 has two lists of cross streets for the two intersections delimiting the block face containing the input address. Like the cross street lists in Function 3's WA2, Function 1's lists are arranged so that the numerically smallest PB5SC in each list is that list's first entry. (However, unlike Function 3's lists, either or both of Function 1's lists can be empty.) Applications can identify an input address to a street stretch by using Function 1's cross street lists to create a stretch-type key in the same way as described above for Function 3. This is useful in some applications that process geographically heterogeneous input data, with some input records, for example, identified by an address and others by an 'on' street and two cross streets. An example of such an application is the New York City Department of Transportation's Street Light Information and Complaints System, which generates and tracks work orders for street light repair work. Among the ways in which this application improves the efficiency of those operations is by providing a means to consolidate all transactions involving street lights located on the same street segment into a single work order, regardless of whether the initial identification of the location is by an address or by an 'on' street and two cross streets.

VII.5 Block Faces: Function 3C

In some applications, data are related to block faces, which are specific to a side of a street, rather than to street stretches or segments, which comprise both sides of the 'on' street. One way in which users commonly specify block faces is in terms of an 'on' street, two cross streets and a compass direction designating the side of the 'on' street, for example: "in Manhattan, the east side of Madison Avenue between East 50th and East 51st Streets". Given such a block face specification, Function 3C can be used to obtain information specific to that block face. (For Manhattan only, the compass orientations of block faces are shifted 30 degrees counterclockwise, to conform to the widespread conventional treatment of the avenues and streets in midtown Manhattan as if they were oriented due north-south and due east-west, respectively. For more details on this 30-degree shift, see the description of Segment Orientation in Appendix 3.)

Function 3C accepts as input the long block faces formed by T-intersections. The sides of stretches opposite to such long block faces consist of more than one block face and are not accepted as Function 3C input.

The input items to Function 3C are an 'on' street, two cross streets and a compass direction specifying the side of the street. The input cross streets, but not the 'on' street, may be pseudo-streets or intersection names. Note that, for Function 3C, the input compass direction has a different significance than it does for Function 2. In the case of Function 3C, the compass direction identifies which side of the street is to be processed. In the case of Function 2, it identifies, for a pair of input streets that intersect at two distinct locations, which of those two intersections is to be processed.

The same combinations of an 'on' street and two cross streets that Function 3 rejects as ambiguous input data are also rejected as ambiguous input data by Function 3C.

When Function 3C is called using two work areas, it returns in its WA2 those Function 3 WA2 items that are not associated with a specific side of the street, such as the lists of cross streets and the Segment Length. It also returns all of those Function 3 WA2 items that are specific to the side of the street specified by the input compass direction, such as the low and high house numbers, zip code and community district for that side of the street.

A successful two-work-area call to Function 3C signifies the following:

- The input 'on' street and two cross streets (or intersection names) specify a street stretch that is valid as Function 3 input (i.e., it is either a single street segment or a multi-segment stretch at least one side of which is a single entire block face)

- The input compass direction is a valid specification (as defined below) of a side of that stretch

- The specified side of the stretch is a single entire block face

As the second condition above implies, Function 3C treats some compass directions as invalid side-of-street specifications for some street stretches. The validity of a compass direction as a specification of a side of a street is determined by the orientation of the segment (or of the last segment of the stretch, if it consists of more than one segment) with respect to the points of the compass, as indicated by the value of the Segment Orientation. If the segment is oriented 'nearly' (i.e., within ten degrees of) due east-west, as indicated by a Segment Orientation value of 'E' or 'W', then its sides can only be validly described as the north and south sides, and Function 3C will reject east and west as side-of-street specifications. Similarly, if the segment is within ten degrees of due north-south, as indicated by a Segment Orientation value of 'N' or 'S' , then it is considered to have only east and west sides, and Function 3C will reject north and south as side-of-street specifications. (Note: recall that all Segment Orientation values in Manhattan are rotated 30 degrees counterclockwise. This has the effect, for example, of causing Geosupport to treat Third Avenue in Midtown Manhattan as a 'nearly' north-south street. Thus, Function 3C accepts as input either the east or west side of Third Avenue between, say, East 50 Street and East 51 Street, but it rejects the north and south sides.)

Of course, most segments are 'diagonal' (not oriented within ten degrees of due north-south or due east-west), in which case all four compass directions are accepted as valid side-of-street specifications. For example, if the segment is oriented northwest-southeast, one side of the segment is simultaneously the north side and the east side, and the other side is simultaneously the south side and the west side.

Consider the example illustrated in Figure VII-15. The east side of Fifth Avenue between East 40th and East 41st Streets is a valid block face specification and is accepted by Function 3C. The same is true for the east side of Fifth Avenue between East 41st and East 42nd Streets. The west side of Fifth Avenue between East 40th and East 42nd Streets is likewise a valid block face specification, in this case designating the long block face of a T-intersection. The east side of Fifth Avenue between East 40th and East 42nd Streets is not a valid block face specification, and is rejected by Function 3C (even though that combination of 'on' street and two cross streets is accepted by Function 3), since that side of Fifth Avenue between those cross streets consists of two block faces.

VII.6 Street Stretches: Function 3S

Function 3S processes street stretches. An input stretch is specified by an 'on' street and (optionally) any two cross streets, using the same WA1 input fields as are used for Function 3. If the 'on' street intersects an input cross street twice, an input compass direction must also be specified to identify which of those two intersections is intended to delimit the stretch. Input cross streets, but not the 'on' street, may be pseudo-streets or intersection names. If no input cross streets are specified, the delimiting nodes of the input stretch default to the 'on' street's beginning and ending nodes, and the input stretch consists of the entire 'on' street.

Function 3S's WA2 contains a list of all of the input stretch's 'intersections' in sequence between the beginning and ending delimiting nodes of the stretch. An 'intersection' can either be a node (as defined in Section VII.2), or it can be a non-specifiable intersection, that is, a point at which the street intersects only with a geographic feature to which a street code has not been assigned.

WA2 also contains a list counter containing the number of such intersections. There is space in the list for a maximum of 350 intersections. Each entry in the list contains the numerically smallest and second smallest PB5SCs for MSW and up to 5 B7SCs for COW of all of the cross streets (not including the 'on' street) at the represented intersection, if any. If there is only one cross street at an intersection, the list entry contains packed zeros in the second cross street field for for MSW and blanks for COW . List entries representing non-specifiable intersections contain packed zeros in both cross street fields for MSW and blanks for COW.

There is also a field in each list entry for a Gap Flag. A 'G' in the Gap Flag indicates that there is a gap in the 'on' street between the node represented by this list entry and the node represented by the previous list entry. In other words, it indicates that there is no segment of the 'on' street connecting those two nodes.

Each list entry also contains the distance in feet between the node represented by this list entry and the node represented by the previous list entry. The maximum value this field can contain is 99,999. Please note that the distance information is only a rough approximation and cannot be used for applications that require precise distance measurements.

In the COW, there is also a Marble Hill/Rikers Island flag, the node id of the intersection and a count of the number of streets at the intersection.

VII.7 Borough Boundary Processing (Functions 2, 3 and 3C)

All of the street configuration functions other than Function 3S allow users to specify locations that lie along a boundary of two boroughs in terms of streets from both boroughs. Function 2, for example, accepts the intersection of Brooklyn's Ridgewood Avenue and Queens's Rockaway Boulevard as a valid input street intersection lying on the Brooklyn-Queens boundary. A more unusual example that Function 2 also accepts as a valid input intersection is the intersection of Atlantic Avenue in Brooklyn and Atlantic Avenue in Queens. Although physically, Atlantic Avenue is a single continuous street that crosses the Brooklyn-Queens border; Geosupport treats the portions of Atlantic Avenue in the two boroughs as two different streets, and therefore recognizes their meeting point at the borough boundary as an intersection.

A borough boundary location can be specified in terms of streets from different boroughs as follows. For street input data that are in the form of street names, there are three WA1 input fields for borough codes called Borough Code 1, Borough Code 2 and Borough Code 3 (see the WA1 layout in Appendix 2). These fields correspond respectively to the three WA1 input street name fields called Street Name 1, Street Name 2 and Street Name 3. A value is always required in Borough Code 1. If no values are loaded into Borough

Code 2 and/or Borough Code 3, the default values are the value in Borough Code 1. When not all of the input street names are in the same borough, the proper value(s) must be inserted into Borough Code 2 and/or Borough Code 3, as appropriate.

If the street input data are in the form of street codes, either as PB5SCs or as B10SCs, each input street code field contains a borough code in its first byte position. This makes it possible to specify input streets from different boroughs using street code input.

The borough boundary processing feature described in this section is not implemented for Function 3S, which requires all three input streets to be from the same borough. If the input streets are in the form of street names, the borough must be specified in the WA1 input field Borough Code 1; Function 3S ignores the contents of the WA1 input fields Borough Code 2 and Borough Code 3.

CHAPTER VIII: THE GEOSUPPORT API - USER PROGRAM CODING AND JCL

VIII.1 Introduction

The Geosupport System's Application Programming Interface (API), the mechanism by which a user-written batch or CICS application program interfaces with the Geosupport System, was described in broad terms in Chapter II. The present chapter describes in detail, for the programming languages most widely used to develop applications on city mainframes, the statements that the user must code in an application program to access Geosupport via the API. The languages covered are COBOL, Assembler (i.e., any dialect of IBM mainframe assembler language), PL/1, C (supported on the mainframe at the Department of Information Technology and Communications (DoITT) by the IBM/C compiler) and NATURAL (a proprietary programming language used with the ADABAS data base management system). An important optional user programming aid, the Geosupport COPY facility, is also discussed. The Geosupport-related JCL that the user must code to compile, link and (for batch applications) execute an application program is also described.

Notes for non-DoITT mainframe users:

- All JCL documented in this chapter is valid for the DoITT mainframe. Variations from this JCL are possible at other data centers where Geosupport is installed, for a variety of reasons. For example, the DSNs of the Geosupport files may differ from those at DoITT to conform to local file naming standards. Variations from DoITT may also be caused by software environment differences, such as the version of the operating system that is running. In addition, certain software products mentioned in this chapter, such as IBM/C or ADABAS, that are installed at DoITT may be unavailable at other data centers. Non-DoITT users should refer any Geosupport-related JCL questions or problems to their data center's Geosupport System Administrator (the system programmer at the data center who is responsible for installing new Geosupport releases).

- Non-DoITT users should also be aware that certain Geosupport files and functions that are available at DoITT may not currently be installed at their data center. Specifically, a foreground file named PAD, which is accessed only by Functions 1A, BL and BN, is relatively large and therefore is not installed at some data centers at which there are no current applications that require Functions 1A, BL or BN. At those data centers, Functions 1A, BL and BN are not available for use. If future applications at those data centers require Functions 1A, BL and/or BN, the PAD file can then be installed, thereby activating those functions.

VIII.2 Review of the Geosupport API

The Geosupport API consists of the following elements:

- Driver: A Geosupport program called the driver that serves as an intermediary between the user's application program and the Geosupport foreground software. The driver exists in the form of a load module, which the user must link-edit with the application program. (The link-editing is performed automatically for NATURAL programs.)

- Work Areas: One or two standard-layout work areas that are used to pass data back and forth between the application program and Geosupport. The user must include the Geosupport work area(s) in the application program.

- Programming Statements: Programming statements that the user must code in the application program to utilize the driver and work area(s) to interface with Geosupport.

The work areas and required programming statements are identical in the batch and CICS environments, except that there is a different driver for each environment. The driver for batch applications is named GBI. The driver for CICS applications is named GOAIDRV. The driver serves two purposes:

- It passes execution control from the user's application program to the Geosupport foreground software, which is external to the application program load module.

- It passes the memory address(es) of the work area(s), which are located within the application program, to the Geosupport foreground software, enabling Geosupport to access the work areas.

The user program must include the required Geosupport work area(s) in its working storage (COBOL, Assembler or C), automatic storage (PL/1) or U size buffer (NATURAL). When the application program issues a call to the driver, either one or two work areas (more precisely, their memory addresses) are passed as parameters of the call. The length and layout of Work Area 1 (WA1) are fixed. The length and layout of Work Area 2 (WA2) are determined by the function and, for functions that have the long WA2 option (discussed in Section II.5), by whether that option is specified. The distinction between one-work-area and two-work-area calls is discussed in Section II.4.

For the convenience of users whose application programs are written in COBOL, Assembler, PL/1, C or NATURAL, Geosupport COPY files are maintained. They contain source code descriptions of all of the work area layouts in each programming language. The Geosupport COPY files are discussed in detail in Section VIII.4. The use of the Geosupport COPY facility is optional but strongly recommended.

In batch applications, the user JCL for the execute step must include JOBLIB or STEPLIB DD statements for the load libraries that contain the Geosupport foreground software. Section VIII.8 describes the JCL required for batch execution, and Appendix 8 contains examples.

Important note for CICS applications written in NATURAL:

> In order for Geosupport's CICS driver to be able to pass control to the Geosupport foreground software properly, the driver must determine whether the user program is written in NATURAL. (This is necessary because NATURAL programs make non-standard program calls in the CICS environment. In a standard call, the address of the parameter list is passed in Register 1. Programs written in COBOL, Assembler, PL/1 and C generate standard calls. However, CICS NATURAL programs use Register 1 for a different purpose. Therefore, for NATURAL programs only, the Geosupport CICS driver uses the Transaction Work Area instead of Register 1 to pass the addresses of the work areas to Geosupport.)
>
> The Geosupport CICS driver determines whether the calling program is a NATURAL program by examining an internal Geosupport table that contains the transaction-IDs of all applications written in NATURAL. If the transaction-ID of a CICS NATURAL program is not in that table, the transaction will terminate abnormally when attempting to call Geosupport. At DoITT, the updating of the Geosupport NATURAL transaction-ID table is the responsibility of DoITT staff. Therefore, DoITT users developing new CICS applications written in NATURAL must make a request to the appropriate DoITT staff to enter the new transaction-IDs into the Geosupport table. NATURAL CICS users running at other computer centers should contact GSS.

VIII.3 Coding API Calls

This section describes the source code statements that the user must code in the application program to call the driver. Also described, for PL/1 and C applications, are the statements required to declare the driver as an external entry point. Such a declaration is not required for COBOL, Assembler and NATURAL.

For all programming languages, the driver can be called either with one or with two calling parameters. The first parameter passes the address of Work Area 1 to the driver. If the application program is making a two-work-area call, the second parameter passes the address of Work Area 2 to the driver.

The programming statements to declare and call the driver are shown below in the form that must be coded for batch application programs. For CICS programs, the user must code these statements in the same way, but with the name of the CICS driver, GOAIDRV, in place of the batch driver, GBI.

In a (batch) PL/1 program, the driver must be declared as an external entry point as follows:

```
DCL GBI EXTERNAL ENTRY OPTION (ASM,INTER);
```

In a (batch) IBM/C program, the driver should be declared as follows:

```
#pragma linkage (GBI,OS)
long GBI(void *,... );
```

If ‘WA1’ and ‘WA2’ are the names that the user has given to the work areas within the application program source code, the statement calling the driver would be coded as follows in a (batch) application program:

Language	One-Work-Area Call	Two-Work-Area Call
COBOL	CALL ‘GBI’ USING WA1.	CALL ‘GBI’ USING WA1, WA2.
Assembler	CALL GBI,WA1,VL	CALL GBI,(WA1,WA2),VL
PL/1	CALL GBI (WA1);	CALL GBI (WA1, WA2);
IBM/C	GBI(&WA1);	GBI(&WA1,&WA2);
NATURAL	CALL ‘GBI’ USING WA1	CALL ‘GBI’ USING WA1 WA2

.

VIII.4 The Geosupport COPY Files

This section describes an optional feature of Geosupport, its COPY files. The Geosupport COPY files contain source code layouts of the Geosupport API work areas in the COBOL, Assembler, PL/1, C and NATURAL programming languages. The use of the Geosupport COPY files can greatly facilitate user programming and is strongly recommended for all applications. The Geosupport COPY files are contained in the Geosupport COPY libraries, which are described below.

In this section, basic concepts of COPY files are explained, and the organization of the Geosupport COPY libraries is described. In Section VIII.5, the specific source code statements that users must code in their programs to utilize the Geosupport COPY files are described. Section VIII.6 describes the JCL required to compile a program that utilizes the Geosupport COPY files.

Overview of COPY Files in General

Many programming languages, including COBOL, Assembler, PL/1, C and NATURAL, have a facility for referring, within the source code of a program, to external files (generically referred to as 'COPY files' in this document) containing source code to be inserted into the program at compile time. (C 'COPY files' are usually called header files; NATURAL 'COPY files' are called Local Data Areas (LDAs).) Each programming language has a declarative command ('COPY' in COBOL and Assembler, '%INCLUDE' in PL/1, '#include' in C, 'LOCAL USING' in NATURAL) for referring to such external COPY files. During program compilation, when the compiler encounters such a command, it dynamically retrieves the source code stored in the named COPY file and processes that source code as if it were an integral part of the program source code. The source code retrieved at compile time from the COPY file serves as input to the compiler only; it is not inserted permanently into the user's program source code file. Only the declarative statement that refers to the COPY file is permanently present in the program source code. Note: declarative statements referencing external source code COPY files are not to be confused with external program calls. Declarative statements are directives to the compiler, and are processed at source code compilation time; program calls are executable statements, performed at application execution time.

For COBOL, Assembler, PL/1 and C, COPY files must reside as members of a Partitioned Data Set (PDS) called a COPY library, which must be made accessible to the compiler by coding a SYSLIB DD statement in the JCL for the compile step. In addition, for COBOL and PL/1, an appropriate compiler option must be specified. For NATURAL, COPY files are called Local Data Areas (LDAs) and reside in the system library in each ADABAS data base. The Data Base Administrator (DBA) must modify each NATURAL application's security profile to make the LDAs accessible to the application.

An ideal situation in which to use COPY files is when numerous programs in an application must all describe the same data structure(s). Using this technique, a source code description of each data structure is stored centrally in a COPY library. All programs requiring one or more of the data structures need only contain declarative statement(s) referring to the appropriate member file(s) in the COPY library. This approach insures that all the programs define the given storage layout in exactly the same way, using the same data item names, data types and data lengths. This facilitates application-wide maintenance and debugging. Changes to a data structure need only be made centrally in the COPY file, rather than separately and redundantly in each program.

Of course, the source code in a program that references a COPY file must be written so that it is compatible with the source code in that COPY file. In particular, for fields that are defined in the COPY file and referred to in the program, the program must use the same data names and must assume the same data types and lengths as does the COPY file.

The Geosupport COPY Libraries

There are two Geosupport COPY libraries that collectively contain COBOL, Assembler, PL/1 and C source code COPY files for all of the Geosupport API work area layouts. Geosupport also has a set of NATURAL LDAs for the work area layouts. The use of these facilities can greatly facilitate user application programming. Among the potential productivity benefits are the following:

- Elimination of the need for application programmers to key into their programs lengthy source code descriptions of the Geosupport work area layouts line by line.

- Standardization of Geosupport data item names among the programs in an application, facilitating troubleshooting and the reassignment of programming staff to programs written by others.

- Standardization of the descriptions (data types and lengths) of Geosupport data items in an application, fostering accuracy and compatibility among programs and files.

- Simplification of the updating of programs to reflect changes to Geosupport work area layouts. Each time a program that references the Geosupport COPY libraries is recompiled, the latest versions of the work area layouts are automatically retrieved.

The Geosupport COPY libraries supporting COBOL, Assembler, PL/1 and C applications are two catalogued Partitioned Data Sets (PDSs) named A030.GEO.COPYLIB2 and A030.GEO.COPYLIB.

For NATURAL applications at DoITT, the DoITT Data Base Administration staff is responsible for installing the Geosupport LDAs in the system library (CSCLIB) for each ADABAS data base and for making the LDAs accessible to each application that needs such access by updating the application's profile. <u>It is the user's responsibility to communicate with the appropriate DoITT staff to request such installation and profile updating.</u>

The Geosupport COPY libraries contain a complete set of COPY files for the Geosupport API work areas in COBOL, Assembler, PL/1, C and NATURAL. Each COPY file contains source code descriptions of one or more of the work areas in one of the supported programming languages, as explained below. The Geosupport COPY libraries support both batch and CICS applications.

The MSW Work Area 2s of Functions 1, and 1E, (regular WA2), 2, 3 and 3C all have the same length, 200 bytes. For each of the supported programming languages except C, the layouts of these work areas are stored together in a single COPY file, coded as redefinitions of the same memory area. (In the COBOL files, this is done using REDEFINES. In the Assembler files, it is done using an ORG to reset the Location Counter. In the PL/1 files, it is done using BASED. In the NATURAL LDAs, it is done using REDEFINE.) Except for C, each of the remaining work area layouts has its own COPY file. For C, there is a single COPY file (called a 'header file' in C terminology) containing the layouts of all of the work areas, including WA1.

The COW Work Area 2s are handled in a similar fashion, through they may not all have the same length.

Tables VIII-1 and VIII-2, below, list all of the MSW and COW COPY files respectively for COBOL, Assembler, PL/1, NATURAL and C. The tables indicate the work areas for which each file contains layouts, the lengths of those work areas in bytes, and the name of each file by programming language. Appendix 5 and Appendix 14 contain printouts of the MSW and COW COPY files respectively.

Table VIII-1: MSW COPY Files for COBOL, Assembler, PL/1, C and NATURAL

WORK AREA	FUNCTION(S)	LENGTH (bytes)	COPY File Name				
			COBOL	ASSEMBLER	PL/1	C	NATURAL
WA1	All	884	W1COB	W1BAL	W1PL1	WAC	GEOLW1
WA2	1 (regular WA2), 1E (regular WA2), 2, 3 (regular WA2), 3C	200	W2COB	W2BAL	W2PL1	WAC	GEOLW2
WA2	1 (long WA2), 1E (long WA2), 3 (long WA2)	300	W2COBL	W2BALL	W2PL1L	WAC	GEOLW2L
WA2	1A&BL (regular WA2), BN (*)	939	W2COB1A	W2BAL1A	W2PL11A	WAC	GEOLW21A
WA2	1A&BL (long WA2) (**)	17,683	W2COB1AL	W2BAL1AL	W2PL11AL	WAC	GEOLW2AL
WA2	3S	4,224	W2COB3S	W2BAL3S	W2PL13S	WAC	GEOLW23S

Table VIII-2: COW COPY Files for COBOL, Assembler, PL/1, C and NATURAL

WORK AREA	FUNCTION(S)	LENGTH (bytes)	COPY File Name				
			COBOL	ASSEMBLER	PL/1	C	NATURAL
WA1	All	1,200	P1COB	P1BAL	P1PL1	PAC	GEOLP1
WA2	1, 1E, 3C	300	P2COB	P2BAL	P2PL1	PAC	GEOLP2
WA2	2	200	P2COB	P2BAL	P2PL1	PAC	GEOLP22
WA2	3	450	P2COB	P2BAL	P2PL1	PAC	GEOLP23
WA2	1A&BL (regular WA2), BN (*)	1363	P2COB1A	P2BAL1A	P2PL11A	PAC	GEOLP21A
WA2	1A&BL (long WA2) (**)	17,750	P2COB1AL	P2BAL1A	P2PL11AL	PAC	GEOLP2AL
WA2	3S	19,274	P2COB3S	P2BAL3S	P2PL13S	PAC	GEOLP23S

(*) Functions 1A, BL and BN share a single regular WA2 layout.

(**) Functions 1A and BL share a single long WA2 layout. (Function BN does not have the long WA2 option.)

.

VIII.5 Coding API Calls When Using Geosupport COPY Files

This section describes the source code statements that COBOL, Assembler, PL/1, C and NATURAL users must code in application programs that use Geosupport COPY files. The required statements consist of declarative statements to reference the COPY files (using the file names in Table VIII-1) and statements calling the driver.

In Section VIII.3, the forms of calls to the driver were given using arbitrary data names for the work areas. In programs that do not use Geosupport COPY files, those names are user-selectable. In the present section, the forms of the driver calls are given again, this time with the specific data names that are required for compatibility with the COPY files. COBOL and C are the only supported languages that permit COPY file users to select their own names for the work areas (but not for the fields within the work areas). Prior to each call to the driver, the program must prime Work Area 1 with the input data to be processed by Geosupport, as described in Section II.3.

The declarative statements referencing COPY files cause the compiler to process the source code contained therein as if it were present within the application program's own source code at the point in the program where the declarative statement is located. Application programs need only reference those Geosupport COPY files that are required for the Geosupport function(s) the program actually calls, although referencing other COPY files does no harm.

COBOL Source Code Statements

To reference Geosupport **MSW COPY files,** COBOL programs must contain the appropriate one(s) of the following statements in WORKING-STORAGE:

01 ANY-NAME-FOR-WA1. COPY W1COB.	WA1, all functions
01 ANY-NAME-FOR-WA2. COPY W2COB.	WA2, Functions 1 & 1E & 3 (regular WA2), 2, 3C
01 ANY-NAME-FOR-WA2-L. COPY W2COBL.	WA2, Functions 1& 1E & 3 (long WA2)
01 ANY-NAME-FOR-WA2-1A. COPY W2COB1A.	WA2, Functions 1A & BL (regular WA2), BN
01 ANY-NAME-FOR-WA2-1AL. COPY W2COB1AL.	WA2, Functions 1A & BL (long WA2)
01 ANY-NAME-FOR-WA2-3S. COPY W2COB3S.	WA2, Function 3S

For COBOL programs that will be executed in the batch environment and that use Geosupport COPY files, API calls are coded as follows:

One-work-area calls, all functions:
CALL ‘GBI’ USING ANY-NAME-FOR-WA1.

Two-work-area calls, Functions 1 & 1E & 3 (regular WA2), 2, 3C:
CALL ‘GBI’ USING ANY-NAME-FOR-WA1 ANY-NAME-FOR-WA2.

Two-work-area calls, Functions 1 & 1E & 3 (long WA2):
CALL ‘GBI’ USING ANY-NAME-FOR-WA1 ANY-NAME-FOR-WA2-L.

Two-work-area calls, Functions 1A & BL (regular WA2), BN:
CALL ‘GBI' USING ANY-NAME-FOR-WA1 ANY-NAME-FOR-WA2-1A.

Two-work-area calls, Functions 1A & BL (long WA2):
CALL ‘GBI’ USING ANY-NAME-FOR-WA1 ANY-NAME-FOR-WA2-1AL.

Two-work-area calls, Function 3S:
CALL ‘GBI’ USING ANY-NAME-FOR-WA1 ANY-NAME-FOR-WA2-3S.

CICS programs issue calls as above but with GOAIDRV in place of GBI.

To reference Geosupport **COW COPY files** include the appropriate COPY files whose names begin with P1 and P2 instead of W1 and W2, e.g. P1COB instead of W1COB. Note that in the COW format, there is no long WA2 for Functions 1, 1E, and 3.

Assembler Source Code Statements

To reference Geosupport **MSW COPY Files,** Assembler programs must contain the appropriate one(s) of the following statements:

COPY W1BAL	WA1, all functions
COPY W2BAL	WA2, Functions 1 & 1E & 3 (regular WA2), 2, 3C
COPY W2BALL	WA2, Functions 1 & 1E & 3 (long WA2)
COPY W2BAL1A	WA2, Functions 1A & BL (regular WA2), BN
COPY W2BAL1AL	WA2, Functions 1A & BL (long WA2)
COPY W2BAL3S	WA2, Function 3S

For Assembler programs that will be executed in the batch environment and that use Geosupport COPY files, API calls are coded as follows:

CALL GBI,W1BAL,VL	One-work-area calls, all functions
CALL GBI,(W1BAL,W2BAL),VL	Two-work-area calls, Functions 1 & 1E & 3 (regular WA2), 2, 3C
CALL GBI,(W1BAL,W2BALL),VL	Two-work-area calls, Functions 1 & 1E & 3 (long WA2)
CALL GBI,(W1BAL,W2BAL1A),VL	Two-work-area calls, Functions 1A & BL (regular WA2), BN
CALL GBI,(W1BAL,W2BAL1AL),VL	Two-work-area calls, Functions 1A & BL (long WA2)
CALL GBI,(W1BAL,W2BAL3S),VL	Two-work-area calls, Function 3S

CICS programs issue calls as above but with GOAIDRV in place of GBI.

To reference Geosupport **COW COPY files** include the appropriate COPY files whose names begin with P1 and P2 instead of W1 and W2, e.g. P1BAL instead of W1BAL. Note that in the COW format, there is no long WA2 for Functions 1, 1E, and 3.

PL/1 Source Code Statements

To reference Geosupport **MSW COPY files,** PL/1 programs must contain the appropriate one(s) of the following statements:

%INCLUDE W1PL1;	WA1, all functions
%INCLUDE W2PL1;	WA2, Functions 1 & 1E & 3 (regular WA2), 2, 3C
%INCLUDE W2PL1L;	WA2, Functions 1 & 1E & 3 (long WA2)
%INCLUDE W2PL11A;	WA2, Functions 1A & BL (regular WA2), BN
%INCLUDE W2PL11AL;	WA2, Functions 1A & BL (long WA2)
%INCLUDE W2PL13S;	WA2, Function 3S

For PL/1 programs that will be executed in the batch environment and that use Geosupport COPY files, API calls are coded as follows:

CALL GBI (W1PL1);	One-work-area calls, all functions
CALL GBI (W1PL1,W2PL1);	Two-work-area calls, Functions 1 & 1E, 2, 3 (regular WA2), 3C
CALL GBI (W1PL1,W2PL1L);	Two-work-area calls, Functions 1 & 1E & 3 (long WA2)
CALL GBI (W1PL1,W2PL11A);	Two-work-area calls, Functions 1A & BL (regular WA2), BN
CALL GBI (W1PL1,W2PL11AL);	Two-work-area calls, Functions 1A & BL (long WA2)
CALL GBI (W1PL1,W2PL13S);	Two-work-area calls, Function 3S

CICS programs issue calls as above but with GOAIDRV in place of GBI. (Note: for either a batch or a CICS PL/1 program, the appropriate Geosupport driver (GBI or GOAIDRV respectively) must be declared as an external entry point. See Section VIII.3.)

To reference the Geosupport **COW COPY files** include the appropriate copy files whose names begin with P1 and P2 instead of W1 and W2, e.g. P1PL1 instead of W1PL1. Note that in the COW format, there is no long WA2 for Functions 1, 1E, and 3.

IBM/C Source Code Statements

To reference the Geosupport **MSW COPY file** IBM/C programs must contain the following statement:

```
#include <wac.h>
```

In the following C source code examples, the letter 'L' appears in upper case to facilitate distinguishing it from the numeric character '1'.

The work area layouts must be declared using the typedefs in the Geosupport COPY file. For example:

C_WA1 anyname_wa1;	WA1, all functions
C_WA2_F1 anyname_wa2_f1;	WA2, Functions 1 & 1E (regular WA2)
C_WA2_F1 anyname_wa2_f1L;	WA2, Functions 1 & 1E (long WA2)
C_WA2_F1A anyname_wa2_f1a;	WA2, Functions 1A & BL (regular WA2), BN
C_WA2_F1AL anyname_wa2_f1aL;	WA2, Functions 1A & BL (long WA2)
C_WA2_F2 anyname_wa2_f2;	WA2, Function 2
C_WA2_F3 anyname_wa2_f3;	WA2, Function 3 (regular WA2)
C_WA2_F3L anyname_wa2_f3L;	WA2, Function 3 (long WA2)
C_WA2_F3C anyname_wa2_f3c;	WA2, Function 3C
C_WA2_F3S anyname_wa2_f3s;	WA2, Function 3S

For C programs that will be executed in the batch environment and that use the Geosupport COPY file, API calls are coded as follows:

GBI (&anyname_wa1);	One-work-area calls, all functions.
GBI (&anyname_wa1,&anyname_wa2_f1);	Two-work-area calls, Functions 1 & 1E (regular WA2)
GBI (&anyname_wa1,&anyname_wa2_f1L);	Two-work-area calls, Functions 1 & 1E (long WA2)
GBI (&anyname_wa1,&anyname_wa2_f1a);	Two-work-area calls, Functions 1A & BL (regular WA2), BN
GBI (&anyname_wa1,&anyname_wa2_f1aL);	Two-work-area calls, Functions 1A & BL (long WA2)
GBI (&anyname_wa1,&anyname_wa2_f2);	Two-work-area calls, Function 2
GBI (&anyname_wa1,&anyname_wa2_f3);	Two-work-area calls, Function 3 (regular WA2)
GBI (&anyname_wa1,&anyname_wa2_f3L);	Two-work-area calls, Function 3 (long WA2)
GBI (&anyname_wa1,&anyname_wa2_f3c);	Two-work-area calls, Function 3C
GBI (&anyname_wa1,&anyname_wa2_f3s);	Two-work-area calls, Function 3S

CICS programs issue calls as above but with GOAIDRV in place of GBI. (Note: for either a batch or a CICS C program, the Geosupport driver (GBI or GOAIDRV respectively) must be declared as an external entry point. (See Section VIII.3.)

To reference the Geosupport **COW COPY files** include the PAC COPY file instead of the WAC COPY file. Note that in the COW format, there is no long WA2 for Functions 1, 1E, and 3.

NATURAL Source Code Statements

NATURAL programs reference Geosupport **MSW LDAs** by containing one or more of the following statements in DEFINE DATA:

LOCAL USING GEOLW1	WA1, all functions
LOCAL USING GEOLW2	WA2, Functions 1 & 1E & 3 (regular WA2), 2, 3C
LOCAL USING GEOLW2L	WA2, Functions 1 & 1E & 3 (long WA2)
LOCAL USING GEOLW21A	WA2, Functions 1A & BL (regular WA2), BN
LOCAL USING GEOLW2AL	WA2, Functions 1A & BL (long WA2)
LOCAL USING GEOLW23S	WA2, Function 3S

For NATURAL programs that will be executed in the batch environment and that use the Geosupport COPY files (LDAs), API calls are issued as follows:

CALL ‘GBI’ USING W1NAT	One-work-area calls, all functions
CALL ‘GBI’ USING W1NAT W2NAT	Two-work-area calls, Functions 1 & 1E & 3 (regular WA2), 2, 3C
CALL ‘GBI’ USING W1NAT W2NATL	Two-work-area calls, Functions 1 & 1E & 3 (long WA2)
CALL ‘GBI’ USING W1NAT W2NAT1A	Two-work-area calls, Functions 1A & BL (regular WA2), BN
CALL ‘GBI’ USING W1NAT W2NATAL	Two-work-area calls, Functions 1A & BL (long WA2)
CALL ‘GBI’ USING W1NAT W2NAT3S	Two-work-area calls, Functions 3S

CICS NATURAL programs issue calls as above but with GOAIDRV in place of GBI.

NATURAL programs reference Geosupport **COW LDAs** by containing one or more of the following statements in DEFINE DATA:

LOCAL USING GEOLP1	WA1, all functions
LOCAL USING GEOLP2	WA2, Functions 1 & 1E & 3C
LOCAL USING GEOLP22	WA2, Function 2
LOCAL USING GEOLP23	WA2, Function 3
LOCAL USING GEOLP21A	WA2, Functions 1A & BL (regular WA2), BN
LOCAL USING GEOLP2AL	WA2, Functions 1A & BL (long WA2)
LOCAL USING GEOLP23S	WA2, Function 3S

For NATURAL programs that will be executed in the batch environment and that use the Geosupport COPY files (LDAs), API calls are issued as follows:

CALL ‘GBI’ USING P1NAT	One-work-area calls, all functions
CALL ‘GBI’ USING P1NAT P2NAT	Two-work-area calls, Functions 1 & 1E & 3C
CALL ‘GBI’ USING P1NAT P2NAT2	Two-work-area calls, Function 2
CALL ‘GBI’ USING P1NAT P2NAT3	Two-work-area calls, Function 3
CALL ‘GBI’ USING P1NAT P2NAT1A	Two-work-area calls, Fns 1A & BL (regular WA2), BN
CALL ‘GBI’ USING P1NAT P2NATAL	Two-work-area calls, Functions 1A & BL (long WA2)
CALL ‘GBI’ USING P1NAT P2NAT3S	Two-work-area calls, Functions 3S

CICS NATURAL programs issue calls as above but with GOAIDRV in place of GBI.

VIII.6 JCL for the Compile Step

COBOL, Assembler, PL/1 and C programs that do not reference Geosupport COPY files do not require any Geosupport-related JCL in the compile step. When compiling a COBOL, Assembler, PL/1 or C program that references Geosupport COPY files, the Geosupport COPY libraries must be made accessible to the compiler, as described below. NATURAL programs are compiled in the usual way, with no special user action required to access the Geosupport LDAs.

The Geosupport COPY libraries that support COBOL, Assembler, PL/1 and C are two catalogued files which at DoITT have the DSNs A030.GEO.COPYLIB2 and A030.GEO.COPYLIB. (At other installations, users should verify these DSNs with the data center's Geosupport System Administrator.) The COPY libraries must be concatenated under the DDname SYSLIB in the JCL for the compile step. Since the two libraries have some member names in common, it is essential to concatenate their DD statements in the proper order as shown below.

Assuming that one of the standard IBM catalogued procedures for compiling is being used, the JCL for SYSLIB should be coded as follows:

COBOL:	//COBOL.SYSLIB	DD DSN=A030.GEO.COPYLIB2,DISP=SHR
	//	DD DSN=A030.GEO.COPYLIB,DISP=SHR
ASSEMBLER:	//ASM.SYSLIB	DD DSN=A030.GEO.COPYLIB2,DISP=SHR
	//	DD DSN=A030.GEO.COPYLIB,DISP=SHR
	//	DD DSN=<*name of user macro library*>,DISP=SHR
	//	DD DSN=SYS1.MACLIB,DISP=SHR
PL/1:	//PLI.SYSLIB	DD DSN=A030.GEO.COPYLIB2,DISP=SHR
	//	DD DSN=A030.GEO.COPYLIB,DISP=SHR
IBM/C:	//COMPILE.SYSLIB	DD
	//	DD DSN=A030.GEO.COPYLIB,DISP=SHR

For Assembler applications, care must be taken to insure that the required Assembler macro libraries are concatenated to SYSLIB, as shown.

For C applications, note that the compiler requires access only to COPYLIB, not to COPYLIB2, and furthermore, care must be taken to insure that COPYLIB is concatenated to the IBM/C header file library, rather than overriding it. Overriding is prevented by coding one DD statement with a blank operand field followed by the DD statement for the Geosupport COPY library, as shown.

For COBOL applications, in addition to providing the DD statements for SYSLIB, the appropriate compiler option, LIB, must also be in effect. Since LIB is the default, it does not have to be explicitly specified.

For PL/1 applications being compiled by a compiler other than the IBM Enterprise PL/1 Compiler, in addition to providing the DD statements for SYSLIB, the appropriate compiler option, MACRO or INCLUDE, must also be in effect. If %INCLUDE is the only kind of preprocessor statement in the program, then the INCLUDE option should be used instead of the MACRO option. This will make compilation faster. If other kinds of preprocessor statements are in the program in addition to the %INCLUDE statement(s), then the MACRO option <u>must</u> be used. For example:

```
// EXEC IBMZC,PARM.PLI='MACRO'
              or
// EXEC IBMZC,PARM.PLI='INCLUDE'
```

In the IBM Enterprise PL/1 Compiler, the meaning of the INCLUDE compiler option has changed. As a result, if %INCLUDE is the <u>only</u> kind of preprocessor statement in the program, then <u>no</u> compiler option reflecting that fact should be coded. If other kinds of preprocessor statements are in the program in addition to the %INCLUDE statement(s), then as is the case with other PL/1 compilers, the MACRO option <u>must</u> be used.

VIII.7 JCL for the Linkage Editor Step

In both batch and CICS applications, the Geosupport driver must be link-edited into the user program. For applications written in NATURAL, this is done automatically with no special user action required. For non-NATURAL applications, either batch or CICS, users must link-edit their programs as explained below.

When link-editing a non-NATURAL application, the user must provide in the JCL for the linkage editor step a DD statement for the Geosupport load library containing the driver, as well as an INCLUDE statement in the SYSIN file specifying the driver. The DD statement should be coded as follows ("YOURDDN" may be replaced by any DDname):

```
//YOURDDN   DD   DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
```

For batch programs, the INCLUDE statement in the SYSIN file should be coded as follows:

```
//LKED.SYSIN DD  *
      •
      •
  INCLUDE YOURDDN(GBI)
      •
      •
/*
```

For CICS programs, the INCLUDE statement should be coded as follows:

```
//LKED.SYSIN DD  *
      •
      •
  INCLUDE YOURDDN(GOAIDRV)
      •
      •
/*
```

VIII.8 JCL for the Execute Step (Batch Applications)

For user programs being executed in the batch environment, the user must provide seven megabytes of memory for Geosupport, in addition to the memory required for the user program itself.

Geosupport Software Files

User programs that are executed in the batch environment also require Geosupport-related DD statements in the JCL for the execute step. A STEPLIB or JOBLIB DD statement must be provided to make the Geosupport foreground component batch load module library accessible to the application. As of Version 10.1 of Geosupport, users do this by coding either of the following DD statements:

```
//STEPLIB    DD    DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD    DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
                   or
//JOBLIB     DD    DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD    DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
```

(NATURAL users, please see the important note at the end of this section regarding the STEPLIB or JOBLIB DD statement.)

Geosupport Data Files

As of Version 10.1 of Geosupport, the user no longer has to provide DD statements for the Geosupport data files. In fact, if these DD statements are provided they are ignored by Geosupport. The data set names of the Geosupport data files are stored in a module called DSNAMES. If you need to use a non-standard Geosupport data file, please see your systems programmer.

Note for NATURAL Users:

When using a NATURAL batch execution procedure, care must be taken to insure that the Geosupport load library is concatenated to the two standard NATURAL libraries, rather than overriding them. Overriding is prevented by coding two DD statements with blank operand fields, followed by the DD statement for the Geosupport load library. The following JCL is an example (any stepname may be used):

```
//STEPNM     EXEC    <NATURAL-procname>,REGION=7M
//STEPLIB    DD
//           DD
//           DD      DSN=A030.GEO.SUPPORT PDSE LOADLIB,DISP=SHR
//           DD      DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
```

CHAPTER IX: GEOSUPPORT BATCH ADDRESS TRANSLATOR (GBAT)

IX.1 Introduction

This chapter describes the Geosupport Batch Address Translator (GBAT), the Geosupport System's batch utility program. Users can often satisfy their requirements for batch Geosupport processing without having to write custom programs by processing their files through GBAT.

GBAT can process any user file that meets certain easily satisfied requirements (described in Section IX.4). It can be used to execute any of the Geosupport functions that are available in the batch environment. It can be used to normalize house numbers and street names, to obtain street codes, to validate geographic locations such as addresses and intersections, and to obtain geographic information about such locations such as cross streets, community district, zip code, tax block and tax lot, spatial coordinates or any of the other information that Geosupport provides.

To run GBAT, the user must set up a batch job, and must create a small 'control file' that controls the GBAT execution. Section IX.2 discusses JCL considerations for setting up the batch job. Section IX.3 outlines the processing that GBAT performs and discusses programmed abnormal terminations. The remaining sections of this chapter discuss each of GBAT's six input and output files. Appendices 9, 10, and 12 also pertain to GBAT. Appendices 9 and 12 contain several GBAT tables that are indispensable references for setting up the control file (Tables A9-1, A9-2 and A9-3), interpreting the MSW GBAT output data (Table A9-4), interpreting the COW GBAT output data (Table A12-2), and setting up the JCL (Table A9-5 for MSW format, Table A12-3 for COW format). Appendix 10 contains annotated sample GBAT jobs, including JCL, control files and output listings.

IX.2 JCL Considerations

GBAT makes calls to the Geosupport System via Geosupport's standard API in the same manner as is done by any user-written batch Geosupport application program. To execute GBAT, the user sets up a batch job that invokes a catalogued procedure called GBAT2, which contains all the JCL necessary to support the Geosupport calls. This 'proc' consists of a single step, the stepname of which is also GBAT2. It contains an EXEC statement that executes the GBAT program, a STEPLIB DD statement specifying the program library containing the GBAT and Geosupport load modules, and DD statements for all of the Geosupport foreground files. The user must add DD statements to the GBAT execution step for GBAT's own input and output files. Those files, and their required DDnames, are as follows:

- Input data file, DDname INFILE or INVSAM. Mandatory. Contains the user's geographic information to be processed. Discussed in Section IX.4.

- Input control file, DDname CARDIN. Mandatory. Contains encoded information that describes the user's input data file and specifies GBAT processing options. Discussed in Section IX.5.

- Input alias file, DDname ALIASES. Optional. Allowable if the function being executed accepts street name input. Contains user-defined street name aliases (alternative names and spelling variants) that GBAT is to use to supplement the street names recognized by Geosupport. Discussed in Section IX.6.

- Output file of accepted data, DDname OUTFILE. Optional. Contains one record corresponding to each input data record that is accepted by Geosupport. The record consists of an exact copy of the input data record, followed by data obtained from Geosupport. Discussed in Section IX.7.

- Output file of rejected data, DDname ERRFILE. Mandatory. Contains one record corresponding to each input data record that is rejected by Geosupport. The record consists of the Geosupport Return Code and Reason Code, followed by an exact copy of the input data record. Discussed in Section IX.8.

- Output print file, DDname SYSPRINT. Mandatory. Contains GBAT messages and summary run statistics. Discussed in Section IX.9.

Two of the GBAT files, the input alias file and the output file of accepted data, are optional. GBAT opens these files only if there are certain entries in the control file specifying their use. If an optional file is not used, a DD statement for that file need not be included in the JCL. However, including such a DD statement causes no harm, unless the file it refers to does not exist, which would cause a JCL error.

In the course of execution, certain conditions (described in Section IX.3) may arise that cause GBAT to terminate abnormally. In all cases in which GBAT exits via a programmed abnormal termination, it issues a Condition Code[1] of 12 or greater, and it produces incomplete or no output files. If the user's job contains any steps following the GBAT execution step (the step that invokes the GBAT2 proc) that are dependent on the existence of the output files that GBAT is expected to create, it is advisable to code the COND parameter in the EXEC statements of those steps so that those steps are bypassed if the Condition Code issued by the GBAT execution step is 12 or greater. For example, if GBATSTEP is the stepname of the GBAT execution step, then coding COND=(12,LE,GBATSTEP.GBAT2) in the EXEC statement of a subsequent step will cause that step to be bypassed if 12 is less than or equal to the Condition Code of GBATSTEP.

IX.3 GBAT Processing and Programmed Abnormal Terminations

Before GBAT begins processing the input data file, it first validates the control file and, if appropriate, the alias file. The entire control file is validated for syntax and content, as described in Section IX.5. Regardless of the outcome of that validation, if the function being executed accepts street name input, and the user has provided an alias file, and the user has specified either ALIASES=VAL or ALIASES=YES in the control file, then the contents of the entire alias file are also validated, as described in Section IX.6. For each error encountered during these validations, GBAT issues an appropriate error message but continues the validation processing. Informational and warning messages may also be issued during the control file and alias file validation processing.

After the control file and alias file validation processing is completed, GBAT determines whether to

---

[1] Each step of a batch job running on an IBM mainframe can issue a Condition Code upon termination of the step. By convention, a Condition Code of '00' indicates normal completion of the step, '04' indicates generally normal completion but with a minor condition warranting a warning, and higher values indicate severe problems or errors causing abnormal termination. Condition Codes appear in the SYSPRINT output job log. A JCL parameter, COND, can be coded in the EXEC statement of any job step to cause that step to be bypassed if the Condition Code of a specified previous step satisfies a specified condition.

Note: Condition Codes are sometimes called 'Return Codes', in IBM documentation and elsewhere. Condition Codes are not to be confused with Geosupport Return Codes (GRCs). The Condition Codes discussed in this chapter are issued by the GBAT program, appear in the output job log and can be tested by the COND JCL parameter. GRCs are issued by the Geosupport System and are returned to the calling application (including to GBAT) in Work Area 1; they do not appear in the output job log and are not accessible to COND. GBAT does include the GRC in each record it writes into the output file of rejected data (discussed in Section IX.8).

terminate abnormally or to commence processing the input data file. GBAT terminates abnormally at this point if there has been at least one control file error or, when an alias file has been provided, if there has been at least one alias file error and ALIASES=VAL has been specified. (In contrast, when ALIASES=YES is specified, the alias file is validated and error messages are issued as appropriate, but alias file errors do not cause GBAT to terminate abnormally.) Conditions that cause GBAT to issue informational or warning messages do not trigger abnormal termination.

The Condition Codes issued for programmed abnormal terminations triggered by control file and alias file errors are as follows:

- Condition Code 12: only the control file had errors (or both the control file and the alias file had errors, but ALIASES=YES was specified, so that any alias file errors do not affect the manner in which GBAT terminates).

- Condition Code 13: only the alias file had errors, and ALIASES=VAL was specified.

- Condition Code 14: both files had errors, and ALIASES=VAL was specified.

When no alias file is provided, Condition Code 12 can occur, but not Condition Codes 13 and 14.

If the control file and alias file validation processing is completed normally, GBAT processes the input data file and writes data to the output files. Specifically, GBAT reads each record from the input data file, and uses the geographic information obtained therefrom as the input data for a standard API call to the Geosupport System. If the information is accepted by Geosupport, GBAT writes a record into the output file of accepted data (unless the user has chosen not to create this optional file). If the information is rejected by Geosupport, GBAT writes a record into the output file of rejected data. Both types of output records consist of exact copies of the input data record together with data that GBAT has obtained from Geosupport. At the completion of execution, GBAT writes out a small report (usually less than one page long) of summary run statistics.

In the course of processing the input data file, a condition called a MAXREJECTS violation, discussed in detail below, may arise. If so, GBAT ceases processing the input data, writes out the report of run statistics reflecting the processing that has occurred up to that point, and terminates abnormally with Condition Code 20. Otherwise, GBAT continues processing until all input data records have been processed, writes out the report of run statistics, and then terminates normally with Condition Code 00.

<u>The MAXREJECTS Feature</u> When coding the control file, if the user specifies incorrect record positions for an input data field, it is likely that Geosupport will reject most or all of the input data records. An optional control entry called MAXREJECTS is designed to prevent GBAT, to the extent possible, from wastefully processing an input data file in its entirety when incorrect record positions have been specified in the control file for an input field. The MAXREJECTS feature does this by causing execution to terminate abnormally with Condition Code 20 if a certain number of records at the beginning of the input data file are all rejected by Geosupport for any reason other than an invalid borough code. (The latter exception is designed to prevent a MAXREJECTS termination from occurring inappropriately when a user file has records that intentionally contain blank or otherwise invalid borough codes because those records represent locations outside of New York City.)

The MAXREJECTS control entry is used to specify the number of consecutive rejected records at the beginning of the input data file (ignoring any records rejected for an invalid borough code) that are to trigger a MAXREJECTS termination. For example, the control entry MAXREJECTS=50 directs GBAT to terminate abnormally with Condition Code 20 if every one of the first 50 input data records that are not rejected because of an invalid borough code is rejected for any other reason.

At the user's discretion, warnings can be treated as if they were rejects for the purpose of triggering a MAXREJECTS abnormal termination; see the discussion of the REJECTWARNINGS control entry in Section IX.7.

The MAXREJECTS control entry is optional. If the user does not code a MAXREJECTS control entry, then the value in effect defaults to MAXREJECTS=200. If the user codes MAXREJECTS=NOMAX, the MAXREJECTS feature is turned off; that is, the entire input data file is processed, regardless of how many records at the beginning of the file are rejected.

Coding incorrect input field specifications in the control file tends to make a MAXREJECTS termination likely, but it does not guarantee it. That is because some input data records may contain values in the incorrectly specified field positions that, purely by coincidence, are valid for the intended data item. If there happened to be such a record near the beginning of the input data file, and that record happened to be accepted by Geosupport, that would preclude a MAXREJECTS termination. Conversely, a MAXREJECTS termination can occur even when there are no control file errors. That is because Geosupport may reject all of the input data records that are within the scope of the MAXREJECTS triggering set simply because those particular records happen to contain geographically invalid data.

Note that the completion of a GBAT execution with Condition Code 00 does not by itself signify that no input data records were rejected by Geosupport. It signifies only that no errors were found in the control file nor (if ALIASES=VAL was coded) in the alias file; that a MAXREJECTS violation did not occur; and therefore that all input data records were processed (but not necessarily accepted) by Geosupport.

IX.4 The Input Data File (DDNAME=INFILE or INVSAM)

This mandatory input file contains the user geographic data to be processed by GBAT. In order for GBAT to be able to process a data file, it must satisfy the following requirements:

- The file must be either a sequential file or a VSAM file accessed sequentially. The DDname INFILE is used for sequential files; INVSAM is used for VSAM files. The file can have either fixed or variable length records, but the maximum permissible record length is 32,000 bytes.

- The file must be geographically homogeneous; that is, all of its records must contain the same type of geographic location to be processed. Heterogeneous files, such as a file in which some records contain addresses and others contain intersections, cannot be processed by GBAT. This restriction follows from the fact that, during one execution, GBAT calls the same Geosupport function to process every input record.

- Each data item that serves as an input item must occupy the same field position(s) within every INFILE (INVSAM) record. These field positions are specified in the control file.

IX.5 The Input Control File (DDNAME=CARDIN)

This mandatory input file contains encoded information that controls the GBAT execution, including the Geosupport function being requested, processing options, and the positions of input fields in the input data records. The control file must be provided as a fixed-length file with an LRECL of 80. Users often provide the control file as an in-stream file imbedded in the JCL.

Control File Syntactic Rules The information in the control file is coded in the form of control entries, which must conform to the following syntactic rules:

- Control entries may be coded in any order.
- Each record in the control file may contain one or more control entries. If more than one control entry is coded within the same record, those entries must be separated from each other by at least one blank, and they may be separated by any number of blanks.
- A control entry must not span two records.
- No blanks are permitted within a control entry.
- A control entry consists of a keyword, followed by an equals sign, followed by either a single variable value or a pair of variable values separated by a comma, depending on the keyword, as follows:
    - Keywords other than those specifying the location of a field within the input data records require one variable, and are of the form KEYWORD=V, where V is a variable value specified by the user. For example, the control entry RECTYPE=1E specifies that Function 1E is to be executed during this GBAT run; ‘RECTYPE’ is the keyword in this control entry, and ‘1E’ is the variable value.
    - Most keywords that specify the locations of input data fields require two variables. Such a control entry is of the form KEYWORD=S,L where S and L specify the starting position and length of the input field, respectively. The two variable values must be separated by a comma. For example, the control entry ONSTREET=58,32 specifies that the input street name field starts in position 58 of the input data record and is 32 bytes long.
    - For keywords that specify the locations of input fields for data items of invariant length, coding the length variable is usually optional. For example, a BIN is always a seven-byte item; therefore, if an input BIN field starts in, say, position 29, the control entry specifying that field may be coded as either BIN=29 or BIN=29,7. However, a BBL is always a ten-byte item, but must be explicitly coded as such. See Table A9-2 for default information.

Many of the control entries are optional. GBAT assigns predetermined default values to the variables of all relevant optional control entries that the user has not coded. The default values are suitable for most applications. GBAT issues messages in SYSPRINT informing the user of all such default assignments.

Control File Validation Processing GBAT validates the control file for syntax and, to a certain extent, for content, as described below. GBAT issues an error message in SYSPRINT for each control file error encountered. After completing the validation of the control file, if there have been any errors, GBAT terminates abnormally without processing the input data file and exits with a Condition Code of either 12 or 14 as described in Section IX.3. Certain conditions encountered during control file validation cause warning messages to be issued, but are otherwise ignored and do not cause GBAT to terminate abnormally.

The control file validations include verifying that all of the control entries that are mandatory for the specified Geosupport function have indeed been coded; that the variable values that have been coded in each control entry are valid values for the given keyword; and that the starting position and length that have been specified for each input data field are consistent with the input data file's record length (that is, they do not in combination specify positions beyond the end of the input data record). To do so, GBAT opens the input data file and obtains its LRECL from its Data Set Control Block (DSCB). In the case of a variable length file, the LRECL in the DSCB is the maximum allowable LRECL of the file, as specified by the user when the file was catalogued. Therefore, for a variable length input data file, GBAT can validate only that the starting positions and lengths of input fields coded in the control file are consistent with the **longest possible** input data file record. For a variable length file, it is the user's responsibility to insure that all starting positions and lengths specified in the control file are valid for the **shortest actual** input data file record. If they are not, unpredictable results may ensue.

If the control file contains more than one control entry for the same keyword, the last such control entry is effective and the others are ignored. However, no warning messages are issued indicating the presence of such duplicate keyword entries.

Appendix 9 contains three reference tables that document the full set of control entries. These tables are indispensable references for setting up control files. Tables A9-1 and A9-2 are organized by keyword, and Table A9-3 is organized by Geosupport function. Table A9-1 lists all the control entries along with narrative descriptions of their formats, purposes and usages, and for most of the control entries, citations to sections of the UPG where pertinent topics are discussed in detail. Table A9-2 indicates, for every control entry, the permissible values and the default values of its variables, and the functions for which that control entry may be used. Table A9-3 indicates, for every function, which control entries and combinations of control entries are permissible, mandatory and optional. See also the sample GBAT jobs in Appendix 10 for examples of control files.

An expeditious approach for creating a new GBAT control file is first to ascertain from Table A9-3 which control entries are mandatory and optional for the function to be executed. Tables A9-1 and A9-2 can then be consulted to review those of the control entries with which the user is unfamiliar.

IX.6 The Input Alias File (DDNAME=ALIASES)

For all Geosupport functions that accept street name input, GBAT users have the option to provide a set of user-defined street name aliases (alternative street names and street name spelling variants) in an input alias file. These aliases supplement the set of street names that Geosupport recognizes of its own accord. It is important to note that the user's aliases supplement Geosupport's street names; they do not supersede them. Also, the aliases in an alias file supplement the Geosupport names only temporarily, that is, only during a GBAT execution in which that particular alias file is provided; GBAT does not 'remember' any user-defined aliases that have been supplied in prior GBAT executions.

The alias feature is intended to enable users to customize GBAT execution for a particular data file. This feature is particularly useful for processing a data file that contains a few street names that are misspelled in a consistent manner in many records. By providing just a few entries in an alias file to identify those misspellings with corresponding 'correct' (Geosupport-recognized) spellings, the user may greatly improve the 'hit' rate without having to modify the data file itself. This could be beneficial, for example, if the data file being processed was obtained from an outside source and the user has no software at hand to modify the contents of the file to correct street name misspellings.

To use the alias feature, a control entry containing the keyword ALIASES must be coded as follows:

- ALIASES=VAL directs GBAT to validate the alias file (as described below), and then to process the input data file only if the alias file had no invalid records. If so, then during the processing of the input data file, the user-defined aliases supplement the set of street names that Geosupport recognizes. Records that result in warnings are not considered invalid in this context, and do not prevent the processing of the input data file.

- ALIASES=YES directs GBAT to validate the alias file, and then to process the input data file regardless of whether there were any invalid alias records. During the processing of the input data file, the user-defined aliases that are in the valid alias records supplement the set of street names that Geosupport recognizes, while those in invalid alias records are not used.

If no ALIASES control entry is coded, or if ALIASES=NO is specified, then GBAT performs no alias file processing, even if an ALIASES DD statement appears in the JCL.

If ALIASES=VAL or ALIASES=YES is specified, except for the circumstance discussed in the next paragraph, the user must add a DD statement to the JCL of the GBAT execution step containing the DDname ALIASES, referencing the file that the user wishes to use as the alias file during this GBAT execution.

If ALIASES=VAL or ALIASES=YES is specified, but the function being executed does not accept street name input, then a warning message is issued during control file validation, and the ALIASES control entry, as well as the ALIASES DD statement in the JCL (if any), are otherwise ignored; in particular, no alias file validating is performed in this circumstance.

The alias file must be a sequential file. Although it is expected that most alias files will have at most a few dozen records, GBAT is designed to accommodate alias files of up to 5,000 records. The alias file must have a record length of 80 and must conform to the following layout:

Record Layout of Alias File

Field	**Size**	**Positions**	**Comments**
Borough Code	1	1	Standard Geosupport borough codes
User's Alias Street Name	32	2-33	Need not be in normalized format
Street Name Recognized by Geosupport	32	34-65	Need not be in normalized format
Filler	15	66-80	Blanks

Alias File Validation Processing GBAT validates each record in the alias file, and writes an appropriate message to SYSPRINT for each error or warning condition encountered. A basis of the validation processing is that the alias street name is supposed to be a name that is not already recognized by Geosupport, whereas the street name in the field labeled 'Street Name Recognized by Geosupport' is supposed to be recognized. The alias file validation processing is as follows:

- 'Normal' case: if the alias name is not recognized, and the putative Geosupport-recognized name is in fact recognized, the alias file record is valid.

- Error: If the alias name and the putative Geosupport-recognized name are identical, the alias record is invalid. This condition tends to indicate that the user inadvertently entered the alias name incorrectly when creating this record.

- Warning: If the two names are different, and they are both recognized by Geosupport, and they have the same seven-digit street code (B7SC), then the alias record is superfluous but harmless. A warning message is issued, and the alias name is used.

- Error: If both names are recognized, but they have different B7SC values, the alias record is invalid.

- Error: If the putative Geosupport-recognized name is not in fact recognized, the alias record is invalid.

Note: GBAT does not check whether there is more than one record in the alias file containing the same alias name. If there is more than one, only the first valid record (if any) is used during the processing of the input data file; the other records containing that alias name are validated but are otherwise ignored. It is the user's responsibility to insure that the alias file does not have multiple records containing the same alias name. GBAT issues no warning message indicating the existence of such records.

IX.7 The Output File of Accepted Records (DDNAME=OUTFILE)

This optional output file contains a record corresponding to each input data record accepted by Geosupport. The user can specify whether OUTFILE is to be created, and if so, how its records are to be constituted, using the GEOCODE control entry. The user can specify whether warnings are to be treated as accepted records or as rejects using the REJECTWARNINGS control entry. These control entries are discussed in detail below.

Controlling the Creation and Contents of OUTFILE with GEOCODE GBAT creates either two or three output files, depending on the (coded or default) value in effect for the GEOCODE control entry. If GEOCODE=VAL is explicitly coded (it is never the default), only ERRFILE and SYSPRINT are created. If the value in effect for GEOCODE is other than VAL, then OUTFILE, the file of accepted records, is also created.

The purpose of the option GEOCODE=VAL is to enable the user to validate the input data file while avoiding the execution-time overhead that would be incurred to create OUTFILE. The user can execute GBAT repeatedly with GEOCODE=VAL, each time correcting as many rejected input data records as possible, until the rejection rate is acceptable to the user. At that point, a final execution with GEOCODE=NO, YES or ALL can be run to obtain OUTFILE.

In all cases in which OUTFILE is created, its records are formed by appending data obtained from Geosupport to exact copies of the accepted input data records. The length and layout of the appended Geosupport data depend on the function requested and on the GEOCODE option that is in effect, as described below.

For GEOCODE=NO, the appended items consist generally of output items from Work Area 1 appropriate to the given function. For MSW format, Table A9-4 in Appendix 9 lists, by function, the precise layout of the data appended for GEOCODE=NO. For the COW format, see Table A12-2 in Appendix 12. In general terms, the items that are appended are as follows:

- For functions involving street names, the appended information includes normalized street name(s) and Geosupport street code(s). All normalized street names are provided as 32-byte items, blank-filled on the right as necessary. All street codes are provided as ten-digit street codes without a borough code (10SCs).

- For functions involving house numbers, normalized house numbers are appended. For Functions 1, 1A and 1E, each normalized house number is provided in two formats: the normalized House Number in Display format (HND), a 12-byte item in MSW format, a 16 byte item in COW format, and the normalized House Number in a special format for the Department of Housing Preservation and Development (HNHPD), an 8-byte item (returned with MSW format only). For Functions D, DG and DN, the HNHPD is not returned.

- For Function BL, the 10-byte BBL in standard format is appended. (The standard BBL consists of the borough code, the 5-byte tax block and the 4-byte tax lot.)

- For Function BN, the BIN is appended.

<u>For GEOCODE=YES</u>, each OUTFILE record is formed by appending to a copy of the input record the entire Work Area 2. (See Appendix 2 and Appendix 13 for Geosupport work area layouts, MSW and COW respectively.) GEOCODE=YES is invalid for functions that do not have a WA2. For functions that have the long WA2 option, the long WA2 is appended only if the user has explicitly specified LONGWA2=YES in the control file; if the user specifies LONGWA2=NO, or does not specify a LONGWA2 control entry, then the regular WA2 is appended.

<u>For GEOCODE=ALL</u>, each OUTFILE record is formed by concatenating an exact copy of the input record, followed by the data appended for the given function when GEOCODE=NO is specified (as listed in Table A9-4 in Appendix 9 for MSW and Table A12-2 in Appendix 12 for COW), followed by the data appended when GEOCODE=YES is specified. GEOCODE=ALL is invalid for functions that do not have a WA2.

For MSW, Table A9-5 in Appendix 9 lists the length of the appended data by function and GEOCODE value. For COW, see Table 12-3 in Appendix 12. When setting up the JCL, the user must specify the LRECL parameter in the OUTFILE DD statement to equal the sum of the LRECL of the input data file and the length of the appended data as indicated in Table A9-5 for MSW, and Table 12-3 for COW.

<u>Controlling the Treatment of Warnings with REJECTWARNINGS</u> Every input data record that produces a Geosupport Return Code (GRC) of ‘00’ is treated as an accepted record; that is, the following actions are taken:

- If OUTFILE is being created, GBAT writes a corresponding output record into OUTFILE.

- Regardless of whether or not OUTFILE is being created, the record contributes to the count of accepted records that appears in the SYSPRINT report of run statistics (see Section IX.9)

- If the record is within the scope of the MAXREJECTS triggering set (see Section IX.3), it precludes a MAXREJECTS abnormal termination.

Every input data record that produces a GRC of greater than ‘01’ is treated as a reject; that is, the following actions are taken:

- GBAT writes a corresponding output record into ERRFILE, the output file of rejected records.

- The record contributes to the count of rejected records that appears in the SYSPRINT report of run statistics (see Section IX.9)

- Unless the record is rejected for an invalid borough code, it is counted as a rejected record for the purpose of determining whether a MAXREJECTS abnormal termination is to be triggered.

At the user's discretion, input data records that produce warnings (GRC = '01') either can all be treated as accepted records or they can all be treated as rejects. This choice is specified using the optional REJECTWARNINGS control entry, as follows:

- If REJECTWARNINGS=YES is specified, only the GRC '00' records are treated as accepted records; GRC '01' records are treated as rejects.

- If REJECTWARNINGS=NO is specified, then the GRC '01' records as well as the GRC '00' records are treated as accepted records.

- (Default) If no REJECTWARNINGS control entry is supplied, then the default value is NO; that is, the GRC '01' records as well as the GRC '00' records are treated as accepted records.

IX.8 The Output File of Rejected Records (DDNAME=ERRFILE)

This mandatory output file contains a record for each 'rejected' input data record. The value of the REJECTWARNINGS option that is in effect determines which input data records are treated as rejects, as explained in Section IX.7.

Each ERRFILE record consists of four bytes, followed by an exact copy of the input data record. The four bytes consist of the two-byte GRC, followed by a one-byte filler containing a '-' (dash character) for display readability, followed by the one-byte Reason Code. The LRECL of ERRFILE must always be four greater than that of the input data file. It is the user's responsibility to specify the LRECL of ERRFILE correctly in the JCL.

IX.9 The Output Print File (DDNAME=SYSPRINT)

This mandatory output file contains all GBAT messages, including routine informational messages, abnormal termination messages, control file and alias file validation error messages, and control file default assignment informational messages.

If GBAT terminates normally, or if it terminates abnormally with a MAXREJECTS violation, SYSPRINT also contains a report of run statistics, which is usually less than one page long. The user can specify a title line for the report, consisting of any character string of up to 73 bytes, by using the TITLE control entry.

The report of run statistics indicates the number of input records processed, the number accepted by Geosupport and the number rejected, all itemized by borough. The rejected record statistics are also itemized by GRC. Input data records that result in Geosupport warnings are counted in the report of run statistics either as accepted records or as rejects, depending on the value of the REJECTWARNINGS option that is in effect, as described in Section IX.7. In addition, the report contains a summary list of all the GRCs that have occurred during the given GBAT execution along with their corresponding Geosupport messages.

APPENDICES AND GLOSSARY

APPENDIX 1: GEOSUPPORT FUNCTIONS - QUICK REFERENCE

Introduction

This appendix contains a summary description of each Geosupport function. The entry for each function includes the following elements:

- Description of function and UPG citations: A brief narrative description of the function's purposes, main features and principal output data, with references to salient sections in the body of the UPG. (For a comprehensive list of output data items, see the corresponding work area layouts in Appendix 2.)

- Validation: A description of the validation significance of a successful two-work area call to the function. (The nature of the validation significance of a one-work area call is described in Section II.4.) Entries in this appendix for functions that cannot be called using two work areas do not have a validation section.

- Input fields: A list of mandatory and optional WA1 input fields used to call the function. All input field names are listed in this appendix as they appear in the WA1 layout in Appendix 2 for MSWs and Appendix 13 for COWs, except for street and house number input fields, which are listed as follows:

  Input street fields are usually listed in this appendix generically, using the terms 'Street-1', 'Street-2' and 'Street-3'. Input street data may be in the form of either street names or street codes; input street code data may be in several forms (see Section IV.8). The terms 'Street-1', 'Street-2' and 'Street-3' refer to any of the following three sets of WA1 input fields, the choice of which is at the discretion of the application designer: Street Name-1, Street Name-2 and Street Name-3; or PB5SC-1, PB5SC-2 and PB5SC-3 (MSW only); or B10SC-1, B10SC-2 and B10SC-3. (Note: B5SC-x (where x = 1, 2 or 3) input and B7SC-x input is located left-justified and space-filled in the corresponding B10SC-x input fields.)

  The generic street input field names, 'Street-1', 'Street-2' and 'Street-3', are not used in the entries for Function 1N, which accepts input street names only, and Functions D, DG and DN, which accept input street codes only.

  Input house number fields are listed generically using the term 'House Number'. Input house numbers may be in a displayable, character format, using the WA1 input field called House Number, or they may be in HNI format (MSW, see Section V.2), using the WA1 input field House Number in Internal Format (HNI), or they may be in HNS format (COW, see section V.2) ; using the WA1 input field House Number in Sort Format (HNS).

- Selected Geosupport Return Codes: A list of selected Geosupport Return Codes (GRCs) and Reason Codes that the function can issue, with brief explanations. Only certain notable GRCs specific to the function are included. GRCs that are self-explanatory or that apply to many functions, such as those relating to system errors or to street name normalization and recognition problems, are not included. For a complete list of GRCs, Reason Codes and Messages, see Appendix 4.

Appendix 2 (MSW) and Appendix 13 (COW) contains the work area layouts of all of the Geosupport functions. The abbreviated notation for street code items listed in Table IV-1 (at the end of Chapter IV) is used throughout Appendices 1, 2 and 13.   Below is a summary list of the Geosupport functions.

Summary of Geosupport Functions

Function	User Input Data	Geosupport Output Data
1	Address or Non-Addressable Place name (NAP)	Block face-level data
1A	Address or NAP	Tax lot- and building-level data
1E	Address or NAP	Block face data, political geography
1N	Street Name	Normalized street name, street code
2	Intersection	Intersection-level data
3	Street Segment	Segment-level data
3C	Block Face	Block face-level data
3S	Street Stretch	Stretch-level data
BB	Borough Code + Character String	Up to 10 alphabetized street names
BF	Borough Code + Character String	Up to 10 alphabetized street names
BL	Borough Code+Tax Block+Tax Lot	Tax lot- and building-level data
BN	Building Identification Number	Tax lot- and building-level data
D	Borough Code+5-Digit Street Code and/or HNI/HNS	Primary name of street and/or HND
DG	Borough Code+7-Digit Street Code and/or HNI/HNS	Preferred street name of local group and/or HND
DN	Borough Code+10-Digit Street Code and/or HNI/HNS	Specific street name spelling and/or HND

Function 1

Description: Function 1 processes an input address or input Non-Addressable Place name (NAP) (see Section III-6). When called using two work areas, Function 1 returns information about the block face containing the input address or NAP. This information includes the cross streets at the two intersections delimiting the block face, and a set of geographic district identifiers including zip code, census tract and community district. Function 1 can be called with the long WA2 option.

See Chapter V for a detailed discussion of Function 1 and how it differs from Function 1A.

Validation: A successful outcome of a two-work area call to Function 1 signifies (assuming address rather than NAP input) only that the input address falls within a valid range of addresses of the same parity (odd or even house numbers) allocated to some block face; it does <u>not</u> signify that there is an actual building having the input address. (To validate the latter condition, Function 1A must be used. See Section V.4.)

Input Fields:

Field	Value	Comments
Function Code	'1 ' ('1' followed by a blank)	Required.
Work Area Format Indicator	'C' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	'1'=Manhattan, '2'=Bronx, '3'=Brooklyn, '4'=Queens, '5'=Staten Island	Required.
House Number		Required for address input except free-form addresses (see Section V.3). Not used for NAP input (see Section III.6).
Long WA2 Flag (MSW Only)	'L' = Long WA2, Blank = regular WA2	Optional; default (blank) is regular WA2. See Sec. II.5.
Street-1		Required.
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.

Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.
Cross Street Names Flag	'E' = return names Blank = do not return names	Optional
Roadbed Request Switch	'R' = Roadbed info requested Blank = Generic info requested	Optional; default (blank) requests generic information.

<u>Selected Geosupport Return Codes</u>:

<u>GRC Value/ Reason Code Value</u>	<u>Meaning</u>
01/V	(Warning) The input was a vanity address or a NAP. This message returns the underlying street name when available.
01/E	(Warning) The output address range returned in WA2 is split by a school district boundary. Therefore, the school district value returned in WA2 applies to only a portion of that address range.
01/P	(Warning) The street segment containing the input address is an irregular curve (i.e., it is curved but not as an arc of a circle). No values are returned in the WA2 Spatial Coordinate fields.
07	The input street was specified as a B5SC (or PB5SC) representing a NAP that is the name of a complex. Five-digit street code input is not permitted for the name of a complex. Either the NAP (the name of the complex) must be specified in the input street name field, or its B7SC or B10SC must be specified in the appropriate input street code field.
28	Partial Street name not valid for freeform address.
29	Intersection name cannot be used as "on" Street
41	The input street name is valid but this entire street has no addresses.
42	The input address does not fall within a valid range of addresses for a block face of the input street.

50	The input street name is not valid for the portion of the street where the input house number is located. See Section IV.5.
75	The input address is a 'duplicate address' - i.e., the same address exists at two different locations on the given input street. (Note: this is not a user data error, but an address that is duplicated on this street in reality.) See Section V.6.
89	Long workarea2 option is invalid for COW format for function 1. It is only valid for MSW for this function.

Function 1A

Description: Function 1A processes an input address or input NAP. When successfully called using two work areas, it returns information in WA2 about the tax lot and the building (if any) identified by the input address or NAP. See Chapter VI and particularly Section VI.6.

The information that is returned in WA2 consists of information about the tax lot and the building (if any) identified by the input address or NAP. This information includes the Borough-Block-and-Lot (BBL), which is the Department of Finance's (DOF) identifier for the tax lot; the DOF building class code; the number of buildings on the lot; the number of street frontages of the lot; a flag indicating whether the lot is a condominium; and the Building Identification Number (BIN) (see Section VI.3) of the building identified by the input address, if any. Function 1A can be called with the long WA2 option. The regular WA2 includes a List of Geographic Identifiers (LGI) for the tax lot, including address ranges, BINs and street frontages. The long WA2 includes, instead of the LGI, a List of BINs for all the buildings in the tax lot.

The regular and long WA2s for Function 1A are identical to those for Function BL. Function 1A enables the user to retrieve this information by address, while Function BL enables retrieval by BBL.

Validation: An unconditionally successful outcome of a two-work-area call to Function 1A signifies that a building having the given input address exists. A warning is issued if the input is a pseudo-address (see Section VI.5).

Input Fields:

Field	Value	Comments
Function Code	'1A'	Required.
Work Area Format Indicator	'C ' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	'1'=Manhattan, '2'=Bronx, '3'=Brooklyn, '4'=Queens, '5'=Staten Island	Required.
House Number		Required for address input except free-form addresses (see Section V.3). Not used for NAP input (see Section III.6).
Street-1		Required.
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.
Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.
Long WA2 Flag	'L' = Long WA2, Blank = regular WA2	Optional; default (blank) is regular WA2. See Sections II.5 and VI.6.
1A/BL Version Switch	'S' = standard version, Blank = standard version; valid only for COW	Required for MSW; optional for COW. See Section VI.8.

Selected Geosupport Return Codes:

GRC Value/ Reason Code Value	Meaning
01/8	(Warning) Input address is a pseudo-address.
01/A	(Warning) Function 1A has been called with the regular WA2, but the tax lot identified by the input address or NAP has the List of Geographic Identifiers (LGI) overflow condition, and therefore the LGI in WA2 is incomplete. If a complete list of BINs for the tax lot is required, Function 1A may be called with the long WA2 option for the same input data to retrieve the BINs of all buildings on the tax lot.
04	An invalid value has been specified for the 1A/BL Version Switch. Must be 'S' for standard. See Section VI.8.
07	The input street was specified as a B5SC (or PB5SC) representing a NAP that is the name of a complex. Five-digit street code input is not permitted for the name of a complex. Either the NAP itself (the name of the complex) must be specified in the input street name field, or its B7SC or B10SC must be specified in the appropriate input street code field.
41	The input street name is valid but this entire street has no addresses.
42	The input address is not valid (as defined in Section V.4).
50	The input street name is not valid for the portion of the street where the input house number is located. See Section IV.5.
65	Legacy version of Function 1A is no longer supported. See Technical Bulletin 05-1.
75	The input address is a 'duplicate address' - i.e., the same address exists at two different locations on the given input street. (Note: this is not a user input data error, but an address duplication that exists in reality.) See Section V.6.
90	Invalid value specified for Long WA2 Flag - must be 'L' or blank.

Function 1E

Description: Function 1E processes an input address or input NAP. When called using two work areas, it returns the same WA2 information that is returned by Function 1, and additionally, it returns a set of political districts, including Election, State Assembly and Senate, City Council and Congressional Districts. The layouts of WA2 for Functions 1 and 1E are identical, except for the political district fields, which are fillers in Function 1's WA2. Function 1E reads an additional file to those read by Function 1. Therefore, to avoid unnecessary execution overhead, it is advisable to use Function 1 instead of Function 1E unless the application requires the political districts that only Function 1E provides.

Input Fields: Same as Function 1.

Validation: Same as Function 1.

Selected Geosupport Return Codes: Function 1E's possible GRC values include all of the ones for Function 1, and also the following:

GRC Value/ Reason Code Value	Meaning
01/E	(Warning) The output address range returned in WA2 is split by a school district boundary and/or an election district boundary. Therefore, the school district value and/or the election district value returned in WA2 applies to only a portion of that address range.
56	The input address is associated with more than one Election District (ED). Function 1E requires that this address be specified with a house number suffix to identify a portion of the building specific to one ED. See Section V.4.

Function 1N

Description: Function 1N is used to normalize a street name and obtain its street code. Functions 1, 1A and 1E can do this also, but those functions require an input house number. The purpose of Function 1N is to provide a way to process a street name alone, without a house number. Note that since the input to Function 1N is not a specific location along a street, Function 1N does not perform local street name validation.

Function 1N does not have a Work Area 2, and can only be called using one work area. See Section III.1.

Input Fields:

Field	Value	Comments
Function Code	'1N'	Required.
Work Area Format Indicator	'C ' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12
Borough Code-1	'1'=Manhattan, '2'=Bronx, '3'=Brooklyn, '4'=Queens, '5'=Staten Island	Required.
Street Name-1		Required. (Note: Street code input is not permitted for Function 1N.)
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.
Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.

Selected Geosupport Return Codes: All are self-explanatory.

Function 2

Description: Function 2 processes an input intersection specified either in terms of an intersection name, or in terms of two streets and, when necessary, a compass direction. If the two input streets intersect exactly once, the user should not specify an input compass direction. If the two input streets intersect at two distinct locations, a compass direction must be specified; it serves to identify which of the two intersections the user wishes Geosupport to process. Geosupport does not have the ability to process a pair of input streets that intersect more than twice. Function 2 is discussed in detail in Section VII.2.

When successfully called using two work areas, Function 2 returns information about the input intersection in WA2. If there are more than two streets at an intersection, Function 2 accepts any pair of those streets as user input for that intersection. An intersection that lies on a borough boundary can be specified in terms of one street from each borough, by using the WA1 input field Borough Code 2, as described in Section VII.7.

The information that Function 2 returns in WA2 includes a list of street codes for all streets at the intersection (including the input streets), spatial coordinates for the intersection, and a set of geographic area identifiers including community district, census tract, police precinct and others. If the intersection lies on the boundary of two or more areas of a given type, the identifier for one of those areas is returned.

Validation: A successful outcome of a two-work-area Function 2 call signifies that the two input streets intersect exactly once (if no input compass direction was specified) or exactly twice (if a compass direction was specified). In the double-intersection case, a successful outcome also signifies that the input compass direction is a valid designation of one of the two intersections.

Input Fields:

Field	Value	Comments
Function Code	'2 ' ('2' followed by a blank)	Required.
Work Area Format Indicator	'C' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	'1'=Manhattan, '2'=Bronx, '3'=Brooklyn, '4'=Queens, '5'=Staten Island	Required. Specifies borough of Street Name 1.
Street-1		Required (*). Specifies either of the two streets defining the intersection.
Borough Code-2	(See Borough Code-1)	Optional unless Street Name 2 is in a different borough from Street Name 1; default is value in Borough Code-1.
Street-2		Required (*). Specifies other street defining intersection.
Compass Direction	'N', 'S', 'E' or 'W'	Required only when the two input streets intersect twice, in which case it designates which intersection to process.
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.

Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.
Cross Street Names Flag	'E' = return names Blank = do not return names	Optional.

(*) Note: If either Street 1 or Street 2 contains an intersection name, then the other input street field may either be left blank or it may contain any street that exists at the given intersection.

<u>Selected Geosupport Return Codes</u>:

<u>GRC Value/ Reason Code Value</u>	<u>Meaning</u>
01/H	(Warning) The two input streets intersect only once, but a non-blank input compass direction value has been supplied. The compass direction is superfluous and is ignored. A full complement of output data is returned in the work areas.
02	The two input streets intersect twice, but no input compass direction has been supplied. A valid input compass direction value is required for these input streets.
03	The two input streets intersect more than twice. Geosupport cannot process such intersections.
12	The input information was in the form of an intersection name or a street code of an intersection name. Geosupport recognizes this name or code as valid, but does not yet have this name or code associated with a specific intersection.
30	An input intersection name was specified along with an input street name, but the input street is not part of the intersection.
39	The input compass direction field contains an invalid value, that is, a non-blank value other than 'N', 'S', 'E' or 'W'.
40	The two input streets intersect twice, but the input compass direction value supplied is an invalid descriptor for either of those intersections. If the value supplied is 'E' or 'W', it is invalid because the two intersections are situated approximately due north-south of each other; if the value supplied is 'N' or 'S', it is invalid because the two intersections are situated approximately due east-west of each other.
50	An input street name is not valid for the portion of the street where the input intersection is located. See Section IV.5.

55 At least one of the input streets is a Non-Addressable Place Name (NAP). NAPs are not allowed as input streets for this function.

62 The two input streets do not intersect.

Function 3

Description: Function 3 processes street segments and closely related three-street configurations. A street segment is a part of a street (called the ‘on’ street) between two consecutive cross streets. For example, Madison Avenue (in Manhattan) between East 51st Street and East 52nd Street is a street segment. Madison Avenue between East 51st Street and East 53rd Street is not a street segment, because there is an intervening street, East 52nd Street, between the given cross streets. An exception to the requirement that the input cross streets be consecutive along the ‘on’ street is the case of a T-intersection: Function 3 accepts as input a street configuration that defines the long block face of a T-intersection. (For precise definitions of the terms ‘three-street configuration’, ‘street segment’, and ‘T-intersection’, see Section VII.3.) A street segment intersecting with or lying on a borough boundary can be specified in terms of streets from both boroughs, by using the WA1 input fields Borough Code 2 and Borough Code 3, as described in Section VII.7.

The information returned by a successful two-work-area Function 3 call includes two lists of street codes for all cross streets at the two intersections defined by the input streets; and geographic area codes for the left and right sides of the street, such as the left and right community districts, zip codes, census tracts, etc. ‘Left’ and ‘right’ are defined relative to the ‘on’ street’s ‘logical direction’, which in general is the direction of increasing address. The WA2 information also includes two items called the Segment Azimuth and the Segment Orientation that indicate how the street segment is oriented with respect to the points of the compass. Applications can use either of these items to determine compass direction descriptors for the left and right sides of the street. Another WA2 item, the Cross Street Reversal Flag, can be used to determine left and right relative to the order in which the input cross streets were specified.

Function 3 in MSW format has the Long Work Area 2 Option (see Section II.5).

For a detailed discussion of Function 3, see Section VII.4.

Validation: A successful outcome of a two-work-area call to Function 3 signifies that the input ‘on’ street and two cross streets define a valid street segment or long block face of a T-intersection.

Input Fields:

Field	Value	Comments
Function Code	‘3 ’ (‘3’ followed by a blank)	Required.
Work Area Format Indicator	‘C ‘ = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	‘1’=Manhattan, ‘2’=Bronx, ‘3’=Brooklyn, ‘4’=Queens, ‘5’=Staten Island	Required. Specifies borough of Street Name-1.
Street-1		Required. Specifies ‘on’ street.
Borough Code-2	(See Borough Code-1)	Optional unless borough of Street Name-2 differs from that of Street Name-1. Specifies borough of Street Name-2. Default is Borough Code-1 value.
Street-2		Required. Specifies either cross street.
Borough Code-3	(See Borough Code-1)	Optional unless borough of Street Name-3 differs from that of Street Name-1. Specifies borough of Street Name-3. Default is Borough Code-1 value.
Street-3		Required. Specifies other cross street.
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.
Street Name Normalization Format Flag	‘C’ = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.

Long WA2 Flag (MSW only)	'L' = Long WA2, Blank = regular WA2	Optional; default (blank) is regular WA2. See Section II.5.
Cross Street Names Flag	'E' = return names Blank = do not return names	Optional

<u>Selected Geosupport Return Codes:</u>

<u>GRC Value/ Reason Code Value</u>	<u>Meaning</u>
01/L or R	(Warning) The input 'on' street lies on a borough boundary. The side of street indicated by the Reason Code value is outside of the input borough; no information is returned in WA2 for that side of the street if it is in Nassau or Westchester.
01/Q	These streets involve a dogleg, Shortest Stretch provided. (See section VII.2)
45	Although each of the three input street names was individually recognized, collectively they do not define a valid street segment nor the long block face of a T-intersection.
46	The geographic location specified by the combination of three input streets is ambiguous, i.e., it defines more than one valid segment or T-intersection block face. Geosupport cannot process this input.
50	An input street name is not valid for the portion of the street where the input street segment is located. See Section IV.5.
55	At least one of the input streets is a Non-Addressable Place Name (NAP). NAPs are not allowed as input streets for this function.
89	Long WA2 option is not valid for this function in COW format.
90	Invalid value specified for Long WA2 Flag - must be 'L' or blank.

Function 3C

Description: Function 3C processes block faces specified in terms of an input ‘on’ street, two cross streets and a compass direction designating the side of the street, such as ‘the west side of Madison Avenue between East 53rd Street and East 54th Street’. A block face intersecting with or lying on a borough boundary can be specified in terms of streets from both boroughs, by using the WA1 input fields Borough Code 2 and Borough Code 3, as described in Section VII.7.

When called using two work areas, function 3C returns block face-related information in WA2. This information is a subset of the set of items returned in WA2 by Function 3, consisting of those items that are related to the specified side of the street.

Function 3C is discussed in detail in Section VII.5.

Validation: A successful outcome of a two-work-area call to Function 3C signifies that the input ‘on’ street and two cross streets define a valid street segment or long block face of a T-intersection, and that the input compass direction is a valid designation of a side of this segment. The validity of an input compass direction is determined by the spatial orientation of the segment.

Input Fields:

Field	Value	Comments
Function Code	‘3C’	Required.
Work Area Format Indicator	‘C ‘ = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	‘1’=Manhattan, ‘2’=Bronx, ‘3’=Brooklyn, ‘4’=Queens, ‘5’=Staten Island	Required. Specifies borough of Street-Name-1.
Street-1		Required. Specifies ‘on’ street.
Borough Code-2	(See Borough Code-1)	Optional unless borough of Street Name-2 differs from that of Street Name-1. Specifies borough of Street Name-2. Default is Borough Code-1 value.

Street-2		Required. Specifies either cross street.
Borough Code-3	(See Borough Code-1)	Optional unless borough of Street Name-3 differs from that of Street Name-1. Specifies borough of Street Name-3. Default is Borough Code-1 value.
Street-3		Required. Specifies other cross street.
Compass Direction	'N', 'S', 'E' or 'W'	Required. Specifies side of street of block face (relative to street's 'logical direction' - see Section VII.3).
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.
Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.
Cross Street Names Flag	'E' = return names Blank = do not return names	Optional

<u>Selected Geosupport Return Codes:</u>

Value	Meaning
09	The block face on the side of street specified by the compass direction does not exist in the borough specified for the 'on' street.
39	The input compass direction field contains a non-blank value other than 'N', 'S', 'E' or 'W'.
40	The input compass direction value is invalid as a descriptor of a side of the input street segment, because it is incompatible with the segment's spatial orientation. This condition arises if the segment is oriented approximately east-west and the input compass direction value is specified as 'E' or 'W' (because a street segment oriented approximately east-west has no east and west sides), or the segment is

oriented approximately north-south and the input compass direction value is ‘N’ or ‘S’.

44 Although each of the three input street names was individually recognized, collectively they do not define a valid block face.

46 The combination of these three input streets is ambiguous, i.e., it defines more than one valid block face. Function 3C cannot process such input.

50 An input street name is not valid for the portion of the street where the input block face is located. See Section IV.5.

55 At least one of the input streets is a Non-Addressable Place Name (NAP). NAPs are not allowed as input streets for this function.

Function 3S

Description: Function 3S processes input street stretches. A street stretch is a portion of a street between any two cross streets. If an input cross street intersects with the ‘on’ street twice, an input compass direction is required to identify which of the two intersections is intended. If the user application does not specify input cross streets, Function 3S returns information about the full length of the ‘on’ street. Note that, in a Function 3S call, the input cross streets need not be consecutive along the ‘on’ street.

When successfully called using two work areas, Function 3S returns, in WA2, a list of all intersections in sequence along the ‘on’ street between (and including) the two intersections defined by the input ‘on’ and two cross streets, if any. If the user has not specified input cross streets, the list contains all intersections in sequence from the beginning to the end of the ‘on’ street. The sequence in which the intersections are listed accords with the direction of increasing addresses along the ‘on’ street. Each intersection in the list is specified as a pair of street codes for two of the streets at that intersection. One of the street codes listed for an intersection may or not be the street code of the ‘on’ street.

For each entry in the WA2 list of intersections of the street stretch, there are fields for a distance and a gap flag. The distance is the approximate distance in feet between the given intersection and its predecessor in the list; the gap flag indicates whether the intersection and its predecessor are connected by the ‘on’ street.

Function 3S is discussed in detail in Section VII.6.

Validation: A successful outcome of a two-work-area call to Function 3S using input cross streets signifies that each of the input cross streets intersects the input ‘on’ street.

Input Fields:

Field	Value	Comments
Function Code	'3S'	Required.
Work Area Format Indicator	'C ' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	'1'=Manhattan, '2'=Bronx, '3'=Brooklyn, '4'=Queens, '5'=Staten Island	Required. Specifies borough of Street-1.
Street-1		Required. Specifies 'on' street.
Street-2		Optional. Specifies either cross street.
Compass Direction for First Intersection	'N', 'S', 'E' or 'W'	Required if Street-2 intersects Street-1 ('on'-street) twice. Identifies which of the two intersections is intended.
Street-3		Optional. Specifies other cross street. Must be specified if Street-2 is specified. If Street-2 and Street-3 are not specified, data for full length of street are returned in WA2.
Compass Direction for Second Intersection	'N', 'S', 'E' or 'W'	Required if Street-3 intersects Street-1 ('on'-street) twice. Identifies which of the two intersections is intended.
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.
Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.

Selected Geosupport Return Codes:

Value	Meaning
01/H	(Warning) The input ‘on’ street intersects only once with one of the input cross streets, but a non-blank input compass direction value has been supplied for that intersection. That compass direction is superfluous and is ignored. A full complement of output data is returned in the work areas.
05	A value was supplied in at least one of the input borough code fields other than Borough Code 1. All Function 3S input streets are required to be from the same borough, which must be supplied in the WA1 field Borough Code 1; Borough Code 2 and Borough Code 3 must be blank.
14	The three input streets do not define a street stretch, because the ‘from’ and ‘to’ input intersections are identical.
38	The input ‘on’ street and an input cross street intersect twice, but the input compass direction value supplied is an invalid descriptor for either of those intersections. If the value supplied is ‘E’ or ‘W’, it is invalid because the two intersections are situated approximately due north-south of each other; if the value supplied is ‘N’ or ‘S’, it is invalid because the two intersections are situated approximately due east-west of each other.
39	An input compass direction field contains an invalid value, that is, a non-blank value other than ‘N’, ‘S’, ‘E’ or ‘W’.
55	At least one of the input streets is a Non-Addressable Place Name (NAP). NAPs are not allowed as input streets for this function.
61	Geosupport has no street stretch data for this ‘on’ street. (This condition should never occur for a normal input street. It occurs if the input ‘on’ street is a pseudo-street name (such as DEAD END) or another type of geographic feature that Geosupport recognizes but that Function 3S cannot process as an input ‘on’ street.)
62	The input ‘on’ street does not intersect with one of the input cross streets.
66	The input ‘on’ street intersects with one of the input cross streets more than twice. Function 3S cannot be used to process this combination of input data. (However, Function 3S could be called for this ‘on’ street with no cross streets specified. That call would return data for the full length of the street, including the intersections in question.)

68	The input ‘on’ street intersects with one of the input cross streets twice. An input compass direction value must be supplied to identify which of the two intersections is intended.

Functions BB and BF

Description: Function BB ("browse backward") and BF ("browse forward") enable applications to develop street name browse capability, in order to assist user data entry staff to determine valid spellings of street names that have been rejected. Functions BB and BF can only be called using one work area.

A sequence of repeated calls to Functions BB and/or BF will browse backwards and/or forwards in alphabetical order through the set of all valid normalized street names in a given borough. Each call to one of these functions returns up to ten such names in alphabetical order (or fewer, if there are not ten names remaining in the given borough in the given browse direction). The names are returned in the WA1 field List of Street Names. The number of street names returned in the list is returned in the WA1 field Number of Street Names in List. For COWs, corresponding B7SCs are also returned.

To start a browse sequence, the user application calls either of the browse functions , passing a borough code and character string in the WA1 input fields called Borough Code 1 and Street Name 1, respectively. The input character string can be from one to 32 bytes long. When the last set of ten or fewer names in the given borough is reached, a warning is issued.

For a detailed discussion of Functions BB and BF, see Section III.7.

Input Fields:

Field	Value	Comments
Function Code	‘BB’ or ‘BF’	Required.
Work Area Format Indicator	‘C‘ = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough Code-1	‘1’=Manhattan, ‘2’=Bronx, ‘3’=Brooklyn, ‘4’=Queens, ‘5’=Staten Island	Required.
Street Name-1	Any character string	Required.

Selected Geosupport Return Codes:

GRC Value/ Reason Code Value	Meaning
01/4	(Warning) The last street name has been reached in the specified input borough in the given browse direction. It is possible that fewer than ten street names have been returned in WA1.
97	The input street name is alphabetically beyond the last street name in the specified input borough.

Function BL

Description: Function BL processes a parcel of real property, or tax lot, specified in terms of a standard Department of Finance set of tax lot identifiers, consisting of a combination of a borough code, a tax block number and a tax lot number, collectively called the BBL. When successfully called using two work areas, Function BL returns information about the tax lot in WA2. The WA2 layout for Function BL is identical to that for Function 1A. Function 1A enables the user to retrieve this information by address, while Function BL enables retrieval by BBL.

For a detailed discussion of Function BL, see Chapter VI and particularly Section VI.7.

Validation: A successful outcome of a two-work-area call to Function BL signifies that the input BBL is valid.

Input Fields:

Field	Value	Comments
Function Code	'BL'	Required.
Work Area Format Indicator	'C ' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
Borough-Block-Lot (BBL)		Required.
Long WA2 Flag	'L' = Long WA2, Blank = regular WA2	Optional default (blank) is regular WA2. See Section II.5.

1A/BL Version Switch	'S' = standard version, Blank = standard version; valid only for COW	Required for MSW; optional for COW. See Section VI.8.

Selected Geosupport Return Codes:

GRC Value/ Reason Code Value	Meaning
01/A	(Warning) Function BL has been called with the regular WA2, but the input tax lot has the List of Geographic Identifiers (LGI) overflow condition, and therefore the LGI in WA2 is incomplete. If a complete list of BINs for the tax lot is required, Function BL may be called with the long WA2 option for the same input data to retrieve the BINs of all buildings on the tax lot.
04	An invalid value has been specified for the 1A/BL Version Switch. See Section VI.8.
65	Legacy version of Function BL is no longer supported. See Technical Bulletin 05-1.
90	Invalid value specified for Long WA2 Flag - must be 'L' or blank.

Function BN

Description: Function BN processes a building specified by an input Building Identification Number (BIN). For a discussion of BINs, see Section VI.3.

A successful Function BN call using two work areas returns information about the building in WA2. This includes the building's borough-tax block-tax lot (BBL); a list of geographic identifiers associated with the building (in contrast to Functions 1A and BL, which return geographic identifiers for the entire tax lot, subject to the list's space limitation); a building status flag and date *[not implemented]*; and a condominium flag. Condominiums have unique characteristics discussed in Section VI.4.

Function BN is discussed in detail in Chapter VI and particularly in Section VI.9.

Validation: A successful outcome of a two-work-area call to Function BN signifies that the input BIN is valid.

Input Fields:

Field	Value	Comments
Function Code	'BN'	Required.
Work Area Format Indicator	'C ' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12.
BIN		Required.

Selected Geosupport Return Codes:

GRC Value/ Reason Code Value — Meaning

01/F (Warning) The input BIN is a temporary BIN assigned by GSS to a multi-building tax lot, the individual buildings of which have not yet been assigned permanent BINs. The temporary BIN will be replaced in the future.

20 No input BIN value was specified.

21 The input BIN is not valid.

22 The input BIN has an invalid format: either it contains non-numeric characters, or its first digit is not a valid borough code (the digits 1 through 5), or the digits beyond the first digit are all zeros.

23 The input BIN is a temporary BIN assigned by the NYC Department of Buildings. It exists only in DOB files and is not valid in Geosupport.

Functions D, DG and DN

Description: Functions D, DG and DN are the 'display' functions. They do not actually display anything themselves, but can be used to obtain data items that applications can use to format geographic locations for display on reports, screens, mailing labels etc.

The display functions do not have a WA2 and can only be called using one work area. Each of them can process any combination of up to two input House Numbers in Internal format (HNIs for MSWs) or up to two input House Numbers in Sort format (HNS) for COWs and up to three input street codes. For each successfully processed input HNI/HNS, the corresponding House Number in Display format (HND) is returned in WA1. (House number processing by the display functions is discussed in detail in Section V.2.) For each successfully processed input street code, a corresponding street name is returned in WA1. (Street code processing by the display functions is discussed in Section IV.6.) For each unsuccessfully processed input street code, the corresponding output field is returned containing all question marks. If one input HNI/ HNS is supplied, it may be passed in either input HNI/HNS field. Input street codes may not ‘skip’ any input street code fields.

The display functions process each input item independently of the others, and the input data are not treated as collectively forming a geographic location. In particular, a successful call to a display function does not imply the validation of a geographic location. For example, if there is an input house number and an input street code, these are not treated or validated as forming an address, but are independently processed for conversion to display format. If the input consists of two street codes, these are not treated or validated as forming an intersection, etc.

Functions D, DG and DN differ in the type of street code each processes and in the street name each returns:

- Function D processes input borough-code-and-five-digit street codes, specified either in the form of PB5SCs (MSW) or B5SCs. Input B5SCs are passed left-justified and space-filled in the corresponding WA1 input B10SC fields. For each successfully processed input PB5SC (MSW) or B5SC, Function D returns the corresponding ‘primary’ name for the street (a name from among all of the street’s aliases that GSS has designated as ‘best’ representing the street as a whole).

- Function DG processes input borough-code-and-seven-digit street codes (B7SCs). Input B7SCs are passed left-justified and space-filled in the corresponding WA1 input B10SC fields. For each input B7SC, Function DG returns a street name that GSS has designated as the ‘principal’ street name of the local group of names represented by the given B7SC. Function DG can be used in conjunction with a geographic location-processing function to obtain the ‘preferred street name’ customized for a particular geographic location. (For a discussion of seven-digit street codes and local groups, see Section IV.5.)

- Function DN processes input borough-code-and-ten-digit street codes (B10SCs). For each input B10SC, Function DN returns the unique street name spelling corresponding to it.

Input Fields:

Field	Value	Comments
Function Code	'D ' (D followed by a blank) or 'DG' or 'DN'	Required.
Work Area Format Indicator	'C ' = COW format Blank = MSW format	Optional; default (blank) requests MSW format. See Appendix 12
HNI-1 or HNS-1		Optional.
HNI-2 or HNS-2		Optional.
PB5SC-1 (MSW)		Function D only; optional unless PB5SC-2 is nonblank.
PB5SC-2 (MSW)		Function D only; optional unless PB5SC-3 is nonblank.
PB5SC-3 (MSW)		Function D only; optional.
B10SC-1 (or B5SC-1) (or B7SC-1)		Optional unless B10SC-2 is nonblank. (B5SC-1, B7SC-1 are left-justified, space-filled in B10SC-1)
B10SC-2 (or B5SC-2) (or B7SC-2)		Optional unless B10SC-3 is nonblank. (B5SC-2, B7SC-2 are left-justified, space-filled in B10SC-2)
B10SC-3 (or B5SC-3) (or B7SC-3)		Optional. (B5SC-3, B7SC-3 are left-justified, space-filled in B10SC-3).
SNL	A number between 4 and 32	Optional; default is 32. See Section III.2.
Street Name Normalization Format Flag	'C' = compact format, Blank = sort format	Optional; default (blank) requests sort format. See Section III.3.

Selected Geosupport Return Codes:

GRC Value/ Reason Code Value	Meaning
13/9	At least one input HNI/HNS has a format error. Output HND fields corresponding to unsuccessfully processed input HNIs are returned containing all blanks.
64	At least one input street code is invalid. Output street name fields corresponding to invalid input street code fields are returned containing all '?'.

APPENDIX 2: MAINFRAME-SPECIFIC WORK AREA LAYOUTS (MSW)
(as of Geosupport System Software Version 10.1)

This appendix contains layouts of all of the work MSW areas used with the Geosupport System's API. These layouts are current as of the Geosupport software version indicated above.

Some Geosupport functions can only be called using one work area, Work Area 1 (WA1). Other functions can be called using two work areas, WA1 and Work Area 2 (WA2). For a discussion of one-work-area and two-work-area calls, see Section II.4. WA1 contains both input fields (fields used to pass data from the application to Geosupport) and output fields (fields used to pass data from Geosupport to the application). WA1 is organized so that all the input fields occur first, followed by a filler, followed by all the output fields. WA2 contains output fields only.

All functions use the same WA1 layout, but the set of WA1 fields that are used depends on the function. In the layout of WA1 in this appendix, the column labeled 'Functions' indicates which functions use each field.

The functions that can be called using two work areas use various WA2 layouts of various lengths. In some cases, several functions share a single WA2 layout. For some functions, the user has a choice of two WA2 layouts, a 'regular' WA2 and a 'long' WA2. For a discussion of the long WA2 option, see Section II.5.

The following is a list of all of the Geosupport work areas, indicating the length of each in bytes. Functions that are listed together share a single Work Area 2 layout.

MSW Work Area	Length
WA1, all functions	884
Regular WA2, Function 1	200
Long WA2, Function 1	300
Regular WA2, Functions 1A, BL, BN	939
Long WA2, Functions 1A and BL	17,683
Regular WA2, Function 1E	200
Long WA2, Function 1E	300
WA2, Function 2	200
Regular WA2, Function 3	200
Long WA2, Function 3	300
WA2, Function 3C	200
WA2, Function 3S	4,224

Appendix 3 consists of a data item dictionary describing the fields that occur in the work areas.

Work Area 1 (MSW) Layout for All Functions

Field	Size	Positions	Functions
INPUT FIELDS:			
Geosupport Function Code	2	1-2	All
Borough Code-1[2]	1	3	All but BL, BN, D*
House Number	12	4-15	1, 1A, 1E
House Nr. in Internal Format (HNI)	6	16-21	1, 1A, 1E, D
Street Name-1	32	22-53	All but BL, BN, D*
Street Name-2	32	54-85	2, 3*
Street Name-3	32	86-117	3*
Compass Direction	1	118	2, 3C, 3S
Compass Direction for 2nd Intersection	1	119	3S
PB5SC-1	4	120-123	1, 1A, 1E, 2, 3*, D
PB5SC-2	4	124-127	2, 3*, D
PB5SC-3	4	128-131	3*, D
Roadbed Request Switch	1	132	1, 1E
Borough Code-2	1	133	2, 3, 3C
Borough Code-3	1	134	3, 3C
Street Name Normalization Length Limit (SNL)	2	135-136	All but B*
B10SC-1 (includes B5SC-1 and B7SC-1)	11	137-147	1, 1A, 1E, 2, 3*, D*
B10SC-2 (includes B5SC-2 and B7SC-2)	11	148-158	2, 3*, D*
B10SC-3 (includes B5SC-3 and B7SC-3)	11	159-169	3*, D*
Filler	5	170-174	
Borough-Block-and-Lot (BBL):	10		
Borough	1	175	BL
Tax Block	5	176-180	BL
Tax Lot	4	181-184	BL
Filler	1	185	
Building Identification Number (BIN)	7	186-192	BN
Street Name Normalization Format Flag	1	193	All but B*
Long Work Area 2 Flag	1	194	1, 1A, 1E, 3, BL
Filler - Reserved for Geosupport Use	12	195-206	
HNI-2	6	207-212	D*
Work Area Format Indicator	1	213	All
1ABL Version Switch	1	214	1A, BL
Cross Street Names Flag	1	215	1, 1E, 2, 3, 3C
Filler	4	216-219	

[2] Borough Code values are: '1'= Manhattan, '2'=Bronx, '3'=Brooklyn, '4'=Queens, '5'=Staten Island

Work Area 1 (MSW) Layout for All Functions (continued)

Field	Size	Positions	Functions
OUTPUT FIELDS:			
HND-2	12	220-231	D*
Borough Name	9	232-240	All but D*
Street Name-1Normalized	32	241-272	All but B*
Street Name-2 Normalized	32	273-304	2, 3*, D*
Street Name-3 Normalized	32	305-336	3*, D*
HND	12	337-348	1, 1A, 1E, D*
HNI	6	349-354	1, 1A, 1E
Filler	7	355-361	
PB5SC-1	4	362-365	1*, 2, 3*, D*
Filler	2	366-367	
PB5SC-2	4	368-371	2, 3*, D*
Filler	2	372-373	
PB5SC-3	4	374-377	3*, D*
Attribute Bytes	3	378-380	
Up to ten PB5SCs	40	381-420	BB, BF
B10SC-1	11	421-431	1*, 2, 3*, D*
B10SC-2	11	432-442	2, 3*, D*
B10SC-3	11	443-453	3*, D*
Filler	5	454-458	
BBL Normalized	10	459-468	BL
Reserved	8	469-476	
Street Attribute Indicator	1	477	1*
Reason Code	1	478	All
Filler - Reserved for Geosupport Use	2	479-480	
Geosupport Return Code	2	481-482	All
Geosupport Message	80	483-562	All
Number of Street Names in List (packed)	2	563-564	1*, 2, 3*, BB, BF
List of Street Names: (10 Street Name Fields, 32 Bytes Each)	320	565-884	1*, 2, 3*, BB, BF

***NOTE:**

An asterisk in the second position of a function code is used as a shorthand notation to represent all function codes having the indicated value in the first position, as follows:

1* = 1, 1A, 1E, 1N
3* = 3, 3C, 3S
B* = BB, BF, BL, BN
D* = D, DG, DN

Regular Work Area 2 (MSW) Layout for Function 1

Field	Size	Positions
Filler	22	1-22
Low House Number of Block Face	6	23-28
High House Number of Block Face	6	29-34
Alley/Cross Streets Flag	1	35
Number of Cross Streets at Low Address End	1	36
List of Cross Streets at Low Address End (up to 5 PB5SCs)	20	37-56
Number of Cross Streets at High Address End	1	57
List of Cross Streets at High Address End (up to 5 PB5SCs)	20	58-77
Community District:	3	78-80
Community District Borough Code	1	78
Community District Number	2	79-80
Zip Code	5	81-85
DOT Street Light Contractor Area	1	86
Health Center District	2	87-88
Side of Street Indicator	1	89
Continuous Parity Indicator	1	90
2000 Census Tract	6	91-96
2000 Census Block	4	97-100
Instructional Region (Dept of Education)	2	101-102
Filler	2	103-104
Health Area	4	105-108
Sanitation Recycling Collection Schedule	3	109-111
Feature Type Code	1	112
Interim Assistance Eligibility Indicator	1	113
Curve Flag	1	114
Police Patrol Borough Command	1	115
Police Precinct	3	116-118
Community School District	2	119-120
Filler to Preserve Layout Consistency with WA2 for Function 1E	14	121-134
Filler	1	135
Segment Type Code	1	136
Sanitation District	3	137-139
Sanitation Collection Scheduling Section and Subsection	2	140-141
Fire Division	2	142-143
Fire Battalion	2	144-145
Fire Company Type	1	146
Fire Company Number	3	147-149
Special Address Generated Record Flag	1	150
Reserved for Internal Geosupport Use	1	151
Split Community School District Flag	1	152
DCP-Preferred LGC	2	153-154
LION Face Code	4	155-158

Regular Work Area 2 (MSW) Layout for Function 1 (continued)

Field	**Size**	**Positions**
LION Sequence Number	5	159-163
1990 Census Tract	6	164-169
Filler	4	170-173
Dynamic Block	3	174-176
X Coordinate	7	177-183
Y Coordinate	7	184-190
Segment Length in Feet	5	191-195
Sanitation Regular Collection Schedule	5	196-200

Long Work Area 2 (MSW) Layout for Function 1

Field	Size	Positions
Same as corresponding positions in Function 1's regular WA2	200	1-200
LION Segment-ID	7	201-207
B7SC of True Street	8	208-215
Underlying HNI on True Street	6	216-221
Filler	79	222-300

Regular Work Area 2 (MSW) Layout for Functions 1A, BL, BN

Field	Size	Positions
Filler	28	1-28
Borough-Tax Block-Tax Lot (BBL):	10	29-38
Borough Code	1	29
Tax Block	5	30-34
Tax Lot	4	35-38
Tax Lot Version Number *[not implemented]*	1	39
RPAD Self-Check Code (SCC) for BBL	1	40
Filler	1	41
RPAD Building Classification Code	2	42-43
Corner Code	2	44-45
Filler (reserved)	2	46-47
Number of Street Frontages of Lot	2	48-49
Interior Lot Flag	1	50
Vacant Lot Flag	1	51
Irregularly-Shaped Lot Flag	1	52
Alternative Borough Flag	1	53
Filler	1	54
Strolling Key	13	55-67
List of Geographic Identifiers Overflow Flag	1	68
Reserved for Internal Geosupport Use	1	69
Building Identification Number (BIN) of Input Address or NAP	7	70-76
Condominium Flag	1	77
Condominium Identification Number	4	78-81
Low BBL of this Building's Condominium Units	10	82-91
Filler	1	92
Condominium Billing BBL	10	93-102
Filler	1	103
Condominium Billing BBL SCC	1	104
High BBL of this Building's Condominium Units	10	105-114
Filler	1	115
SBVP (Sanborn Map Identifiers):	8	116-123
Sanborn Borough Code	1	116
Sanborn Volume and Volume Suffix	3	117-119
Sanborn Page and Page Suffix	4	120-123
DCP Commercial Area	5	124-128
Cooperative Identification Number	4	129-132
Filler	4	133-136
Number of Existing Buildings on Lot	4	137-140
Tax Map Identifiers:	9	141-149
Borough Code	1	141
Tax Map Section	2	142-143
Tax Map Volume	2	144-145

Regular Work Area 2 (MSW) Layout for Functions 1A, BL, BN (continued)

Field	Size	Positions
Tax Map Page *[not yet implemented]*	4	146-149
X Coordinate of Internal Label Point	7	150-156
Y Coordinate of Internal Label Point	7	157-163
Filler	18	164-181
Number of Entries in List of Geographic Identifiers	2	182-183
List of Geographic Identifiers, up to 21 entries - each entry consisting of 36 bytes as follows:	756	184-939
Low House Number	6	
Filler	3	
High House Number	6	
Filler	3	
B5SC:		
Borough Code	1	
5-Digit Street Code	5	
DCP-Preferred LGC	2	
BIN	7	
Geographic Identifier Type Code	1	
Filler	1	
Side of Street Indicator	1	

Long Work Area 2 (MSW) Layout for Functions 1A and BL

Field	Size	Positions
Filler	28	1-28
Borough-Tax Block-Tax Lot (BBL):	10	29-38
Borough Code	1	29
Tax Block	5	30-34
Tax Lot	4	35-38
Tax Lot Version Number *[not implemented]*	1	39
RPAD Self-Check Code (SCC) for BBL	1	40
Filler	1	41
RPAD Building Classification Code	2	42-43
Corner Code	2	44-45
Filler (reserved)	2	46-47
Number of Street Frontages of Lot	2	48-49
Interior Lot Flag	1	50
Vacant Lot Flag	1	51
Irregularly-Shaped Lot Flag	1	52
Alternative Borough Flag	1	53
Filler	15	54-68
Reserved for Internal Geosupport Use	1	69
Building Identification Number (BIN) of Input Address or NAP	7	70-76
Condominium Flag	1	77
Condominium Identification Number	4	78-81
Low BBL of this Building's Condominium Units	10	82-91
Filler	1	92
Condominium Billing BBL	10	93-102
Filler	1	103
Condominium Billing BBL SCC	1	104
High BBL of this Building's Condominium Units	10	105-114
Filler	1	115
SBVP (Sanborn Map Identifiers):	8	116-123
Sanborn Borough Code	1	116
Sanborn Volume and Volume Suffix	3	117-119
Sanborn Page and Page Suffix	4	120-123
DCP Commercial Area	5	124-128
Cooperative Identification Number	4	129-132
Filler	4	133-136
Number of Existing Buildings on Lot	4	137-140
Tax Map Identifiers:	9	141-149
Borough Code	1	141
Tax Map Section	2	142-143
Tax Map Volume	2	144-145
Tax Map Page *[not yet implemented]*	4	146-149
X Coordinate of Internal Label Point	7	150-156

Long Work Area 2 (MSW) Layout for Functions 1A and BL (continued)

Field	Size	Positions
Y Coordinate of Internal Label Point	7	157-163
Filler	16	164-179
Number of Buildings on Tax Lot (Maximum = 2,500)	4	180-183
List of Buildings on Tax Lot (each represented by a 7-Byte BIN)	17,500	184-17,683

Regular Work Area 2 (MSW) Layout for Function 1E

Field	Size	Positions
Filler	22	1-22
Low House Number of Block Face	6	23-28
High House Number of Block Face	6	29-34
Alley/Cross Streets Flag	1	35
Number of Cross Streets at Low Address End	1	36
List of Cross Streets at Low Address End (up to 5 PB5SCs)	20	37-56
Number of Cross Streets at High Address End	1	57
List of Cross Streets at High Address End (up to 5 PB5SCs)	20	58-77
Community District	3	78-80
Community District Borough Code	1	78
Community District Number	2	79-80
Zip Code	5	81-85
DOT Street Light Contractor Area	1	86
Health Center District	2	87-88
Side of Street Indicator	1	89
Continuous Parity Indicator	1	90
2000 Census Tract	6	91-96
2000 Census Block	4	97-100
Instructional Region (Dept of Education)	2	101-102
Filler	2	103-104
Health Area	4	105-108
Sanitation Recycling Collection Schedule	3	109-111
Feature Type Code	1	112
Interim Assistance Eligibility Indicator	1	113
Curve Flag	1	114
Police Patrol Borough Command	1	115
Police Precinct	3	116-118
Community School District	2	119-120
Election District	3	121-123
Assembly District	2	124-125
Split Election District Flag	1	126
Congressional District	2	127-128
State Senatorial District	2	129-130
Civil Court District	2	131-132
City Council District	2	133-134
Filler	1	135
Segment Type Code	1	136
Sanitation District	3	137-139
Sanitation Collection Scheduling Section and Subsection	2	140-141
Fire Division	2	142-143
Fire Battalion	2	144-145
Fire Company Type	1	146

Regular Work Area 2 (MSW) Layout for Function 1E (continued)

Field	**Size**	**Positions**
Fire Company Number	3	147-149
Special Address Generated Record Flag	1	150
Reserved for Internal Geosupport Use	1	151
Split Community School District Flag	1	152
Board of Elections-Preferred LGC	2	153-154
LION Face Code	4	155-158
LION Sequence Number	5	159-163
1990 Census Tract	6	164-169
Filler	4	170-173
Dynamic Block	3	174-176
X Coordinate	7	177-183
Y Coordinate	7	184-190
Segment Length in Feet	5	191-195
Sanitation Regular Collection Schedule	5	196-200

Long Work Area 2 (MSW) Layout for Function 1E

Field	Size	Positions
Same as corresponding positions in Function 1E's regular WA2	200	1-200
LION Segment-ID	7	201-207
B7SC of True Street	8	208-215
Underlying HNI on True Street	6	216-221
Filler	79	222-300

Work Area 2 (MSW) Layout for Function 2

Field	Size	Positions
Filler	31	1-31
DCP-Preferred LGC for Street 1	2	32-33
DCP-Preferred LGC for Street 2	2	34-35
Number of Intersecting Streets	1	36
List of Intersecting Streets (up to five PB5SCs, 4 bytes each)	20	37-56
Compass Direction for Intersection Key	1	57
Filler	10	58-67
Instructional Region (Dept of Education)	2	68-69
Fire Division	2	70-71
Fire Battalion	2	72-73
Fire Company Type	1	74
Fire Company Number	3	75-77
Community District	3	78-80
Community District Borough Code	1	78
Community District Number	2	79-80
Zip Code	5	81-85
DOT Street Light Contractor Area	1	86
2000 Census Tract	6	87-92
Filler	3	93-95
Health Area	4	96-99
Filler	9	100-108
LION Node Number	7	109-115
X Coordinate	7	116-122
Y Coordinate	7	123-129
Filler	4	130-133
Police Patrol Borough Command	1	134
Police Precinct	3	135-137
Community School District	2	138-139
Reserved for Internal Geosupport Use	1	140
1990 Census Tract	6	141-146
SBVP1 (Sanborn Map Identifiers):	8	147-154
Sanborn Borough Code	1	147
Sanborn Volume and Volume Suffix	3	148-150
Sanborn Page and Page Suffix	4	151-154
SBVP2 (Sanborn Map Identifiers for Second Map, if any)	8	155-162
Sanborn Borough Code	1	155
Sanborn Volume and Volume Suffix	3	156-158
Sanborn Page and Page Suffix	4	159-162
Filler	38	163-200

Regular Work Area 2 (MSW) Layout for Function 3

Field	Size	Positions
Filler	22	1-22
Curve Flag	1	23
Locational Status	1	24
County Boundary Indicator	1	25
Filler	4	26-29
DCP-Preferred LGC for Street 1	2	30-31
DCP-Preferred LGC for Street 2	2	32-33
DCP-Preferred LGC for Street 3	2	34-35
Number of Cross Streets at Low Address End	1	36
List of Cross Streets at Low Address End (up to five PB5SCs)	20	37-56
Number of Cross Streets at High Address End	1	57
List of Cross Streets at High Address End (up to five PB5SCs)	20	58-77
DOT Street Light Contractor Area	1	78
Cross Street Reversal Flag	1	79
Left Community District	3	80-82
Left Community District Borough Code	1	80
Left Commmunity District Number	2	81-82
Right Community District	3	83-85
Right Community District Borough Code	1	83
Right Community District Number	2	84-85
Left Zip Code	5	86-90
Right Zip Code	5	91-95
Filler	18	96-113
Left Health Area	4	114-117
Right Health Area	4	118-121
Left Instructional Region (Dept of Education)	2	122-123
Right Instructional Region (Dept of Education)	2	124-125
Left Low House Number	7	126-132
Left High House Number	7	133-139
Right Low House Number	7	140-146
Right High House Number	7	147-153
Continuous Parity Indicator	1	154
LION Face Code	4	155-158
LION Sequence Number	5	159-163
Generated Record Flag	1	164
Segment Length in Feet (Packed)	3	165-167
Segment Azimuth	3	168-170
Segment Orientation	1	171
Filler	4	172-175
Left Interim Assistance Eligibility Indicator	1	176
Right Interim Assistance Eligibility Indicator	1	177
Dogleg Flag	1	178

Regular Work Area 2 (MSW) Layout for Function 3 (continued)

Field	Size	Positions
Feature Type Code	1	179
Left Police Patrol Borough Command	1	180
Left Police Precinct	3	181-183
Right Police Patrol Borough Command	1	184
Right Police Precinct	3	185-187
Left Community School District	2	188-189
Right Community School District	2	190-191
Reserved for Internal Geosupport Use	1	192
LION Segment-ID	7	193-199
Segment Type code	1	200

Long Work Area 2 (MSW) Layout for Function 3

Field	Size	Positions
Same as corresponding positions in Function 3's regular WA2	200	1-200
Left 1990 Census Tract	6	201-206
Filler	4	207-210
Left Dynamic Block	3	211-213
Right 1990 Census Tract	6	214-219
Filler	4	220-223
Right Dynamic Block	3	224-226
Left Fire Division	2	227-228
Left Fire Battalion	2	229-230
Left Fire Company Type	1	231
Left Fire Company Number	3	232-234
Right Fire Division	2	235-236
Right Fire Battalion	2	237-238
Right Fire Company Type	1	239-239
Right Fire Company Number	3	240-242
Left 2000 Census Tract	6	243-248
Left 2000 Census Block	4	249-252
Filler	1	253
Right 2000 Census Tract	6	254-259
Right 2000 Census Block	4	260-263
Filler	37	264-300

Work Area 2 (MSW) Layout for Function 3C

Field	Size	Positions
Filler	21	1-21
Curve Flag	1	22
Segment Type Code	1	23
Locational Status	1	24
County Boundary Indicator	1	25
Filler	4	26-29
DCP-Preferred LGC for Street 1	2	30-31
DCP-Preferred LGC for Street 2	2	32-33
DCP-Preferred LGC for Street 3	2	34-35
Number of Cross Streets at Low Address End	1	36
List of Cross Streets at Low Address End (up to 5 PB5SCs)	20	37-56
Number of Cross Streets at High Address End	1	57
List of Cross Streets at High Address End (up to 5 PB5SCs)	20	58-77
Community District	3	78-80
Community District Borough Code	1	78
Community District Number	2	79-80
Zip Code	5	81-85
DOT Street Light Contractor Area	1	86
Filler	7	87-93
2000 Census Tract	6	94-99
2000 Census Block	4	100-103
Filler	1	104
Health Area	4	105-108
Cross Street Reversal Flag	1	109
Side of Street Indicator	1	110
Fire Division	2	111-112
Fire Battalion	2	113-114
Fire Company Type	1	115
Fire Company Number	3	116-118
LION Segment-ID	7	119-125
Low House Number of Block Face	7	126-132
High House Number of Block Face	7	133-139
Alternate Low House Number	7	140-146
Alternate High House Number	7	147-153
Continuous Parity Indicator	1	154
LION Face Code	4	155-158
LION Sequence Number	5	159-163
Generated Record Flag	1	164
Segment Length in Feet (Packed)	3	165-167
Segment Azimuth	3	168-170
Segment Orientation	1	171
Instructional Region (Dept of Education)	2	172-173

Work Area 2 (MSW) Layout for Function 3C (continued)

Field	Size	Positions
Interim Assistance Eligibility Indicator	1	174
Feature Type Code	1	175
Police Patrol Borough Command	1	176
Police Precinct	3	177-179
Community School District	2	180-181
Reserved for Internal Geosupport Use	1	182
1990 Census Tract	6	183-188
Filler	4	189-192
Dynamic Block	3	193-195
Filler	5	196-200

Work Area 2 (MSW) Layout for Function 3S

Field	Size	Positions
Filler	21	1-21
Number of Cross Streets in Stretch (Maximum = 350)	3	22-24
List of Cross Streets in Stretch, each list entry 12 bytes as follows:	4200	25-4224
Smallest PB5SC at Intersection	4	
Second smallest PB5SC at Intersection	4	
Distance from Predecessor in Feet	3	
Gap Flag	1	

APPENDIX 3: DATA ITEM DICTIONARY

This Data Item Dictionary is an alphabetical list of the data items for which there are fields in the Geosupport API work areas, together with descriptive information. In general, data items are listed in this appendix under the names used in the work area layouts in Appendix 2. However, if an item is associated with multiple work area fields having varying field names, and is identical in format and range of values in all those fields, the item is documented in a generically-named entry, and there is also a separate entry for each of those fields consisting only of a reference to the generic entry. For example, there is a generic entry for CENSUS TRACT containing full descriptive information, and there are also entries for 1990 CENSUS TRACT, LEFT 1990 CENSUS TRACT, RIGHT 1990 CENSUS TRACT, 2000 CENSUS TRACT etc., containing only a reference to the entry for CENSUS TRACT.

Each entry consists of an appropriate combination of the following elements:

- Name of Data Item. This might be identical to the name of a specific work area field or it might be a generic name for a data item that is represented by multiple work area fields.

- Field Names. A list of the field names associated with a generic entry.

- Function(s). A list of the Geosupport functions that utilize this data item as either an input or an output item in either WA1 or WA2. If the data item is in the extended portion of WA2 that is passed when the function is called with the long WA2 option, this is so stated. In the list of functions, an asterisk in the second position of a function code is a 'wild card' signifying all functions having the indicated value in the first position, as follows:

  1* = 1, 1A, 1E, 1N
  3* = 3, 3C, 3S
  B* = BB, BF, BL, BN
  D* = D, DG, DN

- Work Area Format: A list of the work area format(s) that apply to this entry, namely, MSW (Mainframe-Specific Work Area) and/or COW (Character-Only Work Area).

- Length and Format. The length of this data item in bytes, and a description of its format, including whether it is numeric, alphabetic or alphanumeric (these terms are defined below); whether it contains any special editing characters; and for numeric items, the justification and the fill character. The following terms and abbreviations are used:

  RJ = Right-Justified
  LJ = Left-Justified
  ZF = Zero-Filled
  BF = Blank-Filled

Numeric:	Contains only the digits 0 through 9, and possibly blanks serving as fill characters only.
Alphabetic:	Contains letters of the alphabet only. LJBF unless otherwise stated.
Alphanumeric:	Can contain any allowable characters, including special characters such as hyphens. LJBF unless otherwise stated.

- <u>Description</u>. A brief narrative description of the data item. The description may include citations to sections of the UPG where the data item is principally discussed. Data items that are self-explanatory have no description and/or citations.

- <u>Valid Values and Code Meanings</u>. The values or ranges of values valid for this data item and, if the item consists of codes, the meaning of each code value.

1A/BL VERSION FLAG - See FUNCTION 1A/BL VERSION FLAG.

1990 CENSUS TRACT - See CENSUS TRACT.

2000 CENSUS BLOCK

Functions:	1, 1E, 3 (MSW: Long WA2, field names LEFT and RIGHT 2000 CENSUS BLOCK), 3 (COW), 3C
Work Area Format:	MSW and COW
Length and Format:	4 bytes. First 3 bytes are numeric. 4th byte contains an alpha suffix (A, B or C).
Description:	Smallest geographic area defined by the U.S. Census Bureau for tabulating the 2000 census. Generally (but not always) corresponds to a physical city block as of 2000. Each 2000 census block is numbered uniquely within its 2000 census tract.

2000 CENSUS TRACT - See CENSUS TRACT.

ALLEY/CROSS STREETS FLAG (ALX)

Functions:	1, 1E
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	Indicates if the segment has been split by alleys, or if the cross streets named in the segment have been copied from a previous or subsequent segment because the segment itself has no cross streets.

Code Value	Meaning
'A'	Split by Alley(s)
'X'	Cross Streets Modified
Blank	Neither Split by Alleys or Cross Streets Modified

ALTERNATE LOW AND HIGH HOUSE NUMBERS

Functions:	3C
Work Area Format:	MSW and COW
Length and Format:	See HOUSE NUMBER
Description:	These two fields are non-blank only if this block face has addresses of both parities (the parity of a number is its attribute of being odd or even). Such a block face is said to have 'continuous parity'. If the block face has continuous parity, the Continuous Parity Indicator is non-blank, the Low and High House Number fields contain the address range for one parity, and the Alternate Low and High House Number fields contain the address range for the other parity. (Which parity is in which set of house number fields is unpredictable.)

ALTERNATIVE BOROUGH FLAG

Functions:	1, 1A, 2, 3, 3C, 3S (COW), BL, BN
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	This flag indicates either that the input address is in Marble Hill or Rikers Island and the alternative (rather than the legal) borough was specified (see Section V.7), or that the input address is on Ruby Street in Brooklyn but it was specified using the alternative (Queens) street name 75 Street (see Section V.8).

Code Value	Meaning
'C'	Ruby Street address specified using 75 Street
'M'	Marble Hill address with Bronx specified
'R'	Rikers Island address with Queens specified

ALX FLAG - See ALLEY/CROSS STREETS FLAG

ANNOTATION POINT - See SPATIAL COORDINATES OF THE INTERNAL LABEL POINT OF THE TAX LOT

ASSEMBLY DISTRICT

Functions: 1E
Work Area Format: MSW and COW
Length and Format: 2 bytes. RJZF
Description: A district of the lower house of the New York State legislature. Consists of an aggregation of Election Districts

ATTRIBUTE BYTE - See STREET ATTRIBUTE INDICATOR

B7SC OF "TRUE" STREET

Functions: 1 and 1E (MSW: Long WA2 only), 1 and 1E (COW)
Work Area Format: MSW and COW
Length and Format: 8 bytes (B7SC)
Description: This item contains the B7SC of the street segment upon which the address specified is actually located. This is the street segment that is identified by the field SEGMENT-ID and by the fields LION FACE CODE and LION SEQUENCE NUMBER. In most cases, the B5SC portion of this item is identical to the B5SC specified in the key. However, the two B5SCs differ when the SPECIAL ADDRESS GENERATED RECORD FLAG is either 'B' or 'V'.

BBL

Functions: 1A, BL, BN
Work Area Format: MSW and COW
Length and Format: 10 bytes in standard version, Numeric. (Note: the legacy version of Functions 1A and BL is no longer supported.)

	Field	Length	Position	Comments
Standard:	Boro	1	1-1	
	Tax Block	5	2-6	RJZF
	Tax Lot	4	7-10	RJZF

Description: The first 6 bytes of the standard BBL consists of the 1-byte borough code followed by the 5-byte tax block field, which contains the tax block value right-justified and zero-filled. The last 4 bytes of the 10-byte standard BBL is the standard tax lot field, which contains the tax lot value right-justified and zero-filled. See Section VI.8.
The BBL ("borough-block-and-lot") identifies a parcel of real property in New York City, called a tax lot. The BBL is composed of the concatenation of the Borough Code, Tax Block and Tax Lot. If the property is a condominium

(indicated by the Condominium Flag), the WA2 BBL field contains the billing BBL of the condominium (see Section VI.4).

BIN - See BUILDING IDENTIFICATION NUMBER

BOARD OF ELECTIONS PREFERRED LGC

Functions:	1E
Work Area Format:	MSW and COW
Length and Format:	2 bytes. RJZF
Description:	This item is the LGC (the sixth and seventh digits of the 10-digit street code) that corresponds to the NYC Board of Elections' preferred street name for a given location.

BOROUGH CODE

Functions:	All functions
Work Area Format:	MSW and COW
Length and Format:	1 byte. Numeric.
Description:	

Code Value	Meaning
1	Manhattan
2	Bronx
3	Brooklyn
4	Queens
5	Staten Island

BOROUGH/BLOCK/LOT - See BBL

BOROUGH/TAX BLOCK/TAX LOT - See BBL

BUILDING IDENTIFICATION NUMBER (BIN)

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	7 bytes. Numeric
Description:	Building Identification Number. A permanent BIN is a seven-digit numerical identifier unique to each building in the City of New York. The first digit is the Borough Code. There are also two types of temporary BINs; those maintained by the Dept. of Buildings (DOB) and those maintained by the Dept. of City Planning (DCP). The temporary BINs assigned by DOB contain the number '8' as the second digit, and the temporary BINs assigned by DCP contain a '9' in the same position. DCP is currently in the process of phasing out all of its temporary BINs.

CENSUS TRACT

Field Names:	1990 CENSUS TRACT, LEFT 1990 CENSUS TRACT, RIGHT 1990 CENSUS TRACT, 2000 CENSUS TRACT, LEFT 2000 CENSUS TRACT, RIGHT 2000 CENSUS TRACT
Functions:	1, 1E, 2, 3 (MSW: Long WA2 Only), 3(COW), 3C

Work Area Format:	MSW and COW
Length and Format:	6 bytes, consisting of numeric 4-digit root followed by numeric 2-digit suffix. The root subfield is RJBF and the suffix subfield is RJZF if any. If the tract number contains no suffix, then the suffix subfield is blank.
Description:	Geographic area defined by the U.S. Census Bureau for the various decennial censuses. Census tracts for a particular census year are numbered uniquely within borough.

CITY COUNCIL DISTRICT

Function:	1E
Work Area Format:	MSW and COW
Length and Format:	2 bytes.
Description:	A district represented by a member of the New York City Council. Consists of an aggregation of Election Districts. There are currently 51 City Council Districts.

CIVIL COURT DISTRICT

Functions:	1E
Work Area Format:	MSW and COW
Length and Format:	2 bytes.
Description:	A district from which a Civil Court judge is elected. Consists of an aggregation of Election Districts.

COMMUNITY DISTRICT

Functions:	1, 1E, 2, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	3 bytes. Numeric. The first byte is the Community District Borough Code, and the second and third bytes are the Community District Number, RJZF.
Description:	There are 59 community districts in the City of New York, as well as 12 Joint Interest Areas (JIAs). The JIAs are major parks and airports that are not contained within any CD.

Code	Meaning
101-112	Manhattan except Marble Hill
164	Central Park
201-212	Bronx except Rikers Island (Note: the Marble Hill section of Manhattan is in Bronx CDs 7 and 8)
226	Van Cortlandt Park
227	Bronx Park
228	Pelham Bay Park
301-318	Brooklyn
355	Prospect Park
356	Brooklyn Gateway National Recreational Area
401-414	Queens (Note: the Rikers Island section of the Bronx is in Queens CD 1)
480	LaGuardia Airport
481	Flushing Meadows - Corona Park
482	Forest Park

483	JFK International Airport
484	Queens Gateway National Recreational Area
501-503	Staten Island
595	Staten Island Gateway National Recreational Area

COMMUNITY SCHOOL DISTRICT

Functions: 1, 1E, 2, 3, 3C
Work Area Format: MSW and COW
Length and Format: 2 bytes
Description: If the block face or the side of the street segment is split between two or more school districts, the corresponding school district field contains 'SP' rather than a valid school district code.

COMPACT FLAG - See STREET NAME NORMALIZATION FORMAT FLAG

COMPASS DIRECTION

Functions: 2, 3C, 3S
Work Area Format: MSW and COW
Length and Format: 1 byte.
Description: In the case of Function 2, the compass direction identifies, for a pair of input streets that intersect at two distinct locations, which of those two intersections is to be processed. (See Section VII.2)
In the case of Function 3C, the compass direction identifies which side of the street is to be processed. (See Section VII.5)
In the case of Function 3S, if the 'on' street intersects the first cross street at two distinct locations, the compass direction identifies which of those two intersections is to be processed. (See Section VII.6)

COMPASS DIRECTION FOR INTERSECTION KEY

Functions: 2
Work Area Format: MSW and COW
Length and Format: 1 byte.
Description: If the first two entries in the LIST OF INTERSECTING STREETS are an instance of the two-node case (i.e., they intersect twice), this field contains a compass direction value identifying the intersection in terms of those two streets. If the two streets are not an instance of the two-node case, this field is blank. If both a 'longitudinal' compass direction ('N' or 'S') and a 'latitudinal' compass direction ('E' or 'W') are valid for this intersection, the longitudinal compass direction value appears in this field.

COMPASS DIRECTION FOR 2nd INTERSECTION

Functions: 3S
Work Area Format: MSW and COW
Length and Format: 1 byte.
Description: If the 'on' street intersects the second cross street at two distinct locations, this compass direction identifies which of those two intersections is to be processed. (See Section VII.6)

CONDOMINIUM FLAG

Functions:	1A, BL, BN	
Work Area Format:	MSW and COW	
Length and Format:	1 byte.	
Description:	<u>Code Value</u>	<u>Meaning</u>
	'C'	Property is a condominium
	Blank	Property is not a condo.

CONDOMINIUM IDENTIFICATION NUMBER

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	4 bytes
Description:	An identification number assigned by the Department of Finance to each condominium in the city. This field is blank for non-condominiums.

CONGRESSIONAL DISTRICT

Function:	1E
Work Area Format:	MSW and COW
Length and Format:	2 bytes.
Description:	A district of the U.S. House of Representatives. Consists of an aggregation of Election Districts.

CONTINUOUS PARITY INDICATOR

Functions:	1, 1E, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	An 'address range' is a sequence of house numbers along an 'on' street between (and including) a Low House Number and a High House Number. Every address range has one of three possible parities: odd, even or continuous. An address range of odd parity consists of all odd house numbers along the 'on' street between the Low and High House Numbers. An even-parity range consists of all even house numbers between the Low and High House Numbers. A continuous-parity range consists of all house numbers (both even and odd) between the Low and High House Numbers. Most New York City block faces contain an address range that is either of even or odd parity. However, some block faces have a continuous-parity address range, usually where the opposite side of the street is non-addressable because it is a park, a body of water, etc. Some examples of the continuous parity case in Manhattan are Central Park West (the east side of the street runs along Central Park and is non-addressable, while the west side has both odd and even addresses); Riverside Drive; and the portion of Fifth Avenue that runs alongside Central Park.

If a New York City block face has a continuous parity address range, Geosupport represents this range as two separate ranges, an odd-parity range and an even-parity range. The practical effect of this depends on the Geosupport function. For Functions 1 and 1E, if an input address lies on a continuous-parity block face, only the range (i.e., the Low and High House Numbers) whose parity is the same as that of the input address is returned in WA2. For Function 3, if an input street segment contains a continuous parity address range, both the odd and the even ranges are returned, in the WA2 fields called Left Low House Number and Left High House Number for the range of one parity, and in the fields Right Low and High House Numbers for the range of the other parity; note that in this case, in reality both the odd and the even ranges are on the same side of the street, even though they are returned in fields called 'left' and 'right'. For Function 3C, if an input block face is on a street segment containing a continuous parity address range (regardless of whether the input block face is on the addressable or the non-addressable side of the segment), both the odd and the even ranges are returned, in the WA2 fields called Low House Number and High House Number for the range of one parity, and in the fields Alternate Low House Number and Alternate High House Number for the range of the other parity.

The field Continuous Parity Indicator indicates, for Functions 1, 1E, 3 and 3C, whether the street segment containing or corresponding to the user input is of the continuous parity type, and if so, which side of the segment is addressable.

Code Value	Meaning
Blank	The street segment does not have a continuous parity address range
'L' or 'R'	The street segment has continuous parity. In this case, the Continuous Parity Indicator indicates which side of the street segment, the left or the right, is addressable. (Left and right are specified with respect to the direction of increasing addresses along the segment)

COOPERATIVE IDENTIFICATION NUMBER

Functions: 1A, BL, BN
Work Area Format: MSW and COW
Length and Format: 4 bytes
Description: This is an identification number assigned by the Department of Finance to each cooperative in the city. This field is blank for non-coops.

CORNER CODE

Functions: 1A, BL, BN
Work Area Format: MSW and COW
Length and Format: 2 bytes
Description:

Code Value	Meaning
'SE','SW','NE','NW'	Tax lot occupies the indicated corner of the physical block
'CR'	Tax lot occupies more than one corner
Blank	Tax lot occupies no corners

COUNTY BOUNDARY INDICATOR

Functions: 3, 3C
Work Area Format: MSW and COW

Length and Format:	1 byte
Description:	This field is non-blank when the street segment lies along a borough boundary. The value of this field indicates which side of the segment is out of borough.

Code Value	Meaning
'L'	Left side of segment is out of borough
'R'	Right side of segment is out of borough
Blank	Neither side is out of borough

CROSS STREET NAMES FLAG

Functions:	1, 1E, 2, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	1 byte character.
Description:	When this flag is set to 'E', the LIST OF STREET NAMES is used to return street names corresponding to the street codes in the LIST OF CROSS STREETS (Functions 1, 1E, 3 and 3C) or in the LIST OF INTERSECTING STREETS (Function 2). The cross street names feature incurs processing overhead, and should only be used when necessary. See LIST OF STREET NAMES for related details.

Code Value	Meaning
Blank	The names of cross or intersecting streets are not returned in the LIST OF STREET NAMES
'E'	The names of cross or intersecting streets are returned in the LIST OF STREET NAMES

CROSS STREET REVERSAL FLAG

Function:	3, 3C
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	This flag indicates the relationship between the order in which the user specified the input cross streets and the direction of increasing addresses along the 'on' street.

Code Value	Meaning
Blank	The direction from Street Name 2 to Street Name 3 (the two input cross street fields) conforms to the direction of increasing addresses
'R'	The direction from Street Name 2 to Street Name 3 is opposite to the direction of increasing addresses

CURVE FLAG

Functions:	1, 1E, 3, 3C
Work Area Format:	MSW and COW

Length and Format:	1 byte character
Description:	This flag indicates whether the given geographic feature segment is in reality curved. If so, the curve may be an arc of a circle or an irregular curve. When the segment specified by the input data is an arc of a circle, Functions 1 and 1E return **Spatial Coordinates** that are positioned relative to this arc rather than to the segment's chord (the imaginary straight line joining the curved feature's endpoints). When the segment specified by the input data is an irregular curve, Functions 1 and 1E return blanks in the **Spatial Coordinate** fields (*q.v.*), and issue a warning with Reason Code value 'P'. In the case of Functions 3 and 3C, if the input data define a street stretch encompassing more than one segment (because of a T-intersection or bend), the Curve Flag is set 'on' (non-blank) if at least one of the constituent segments of the stretch is curved. See also discussion of **Segment Length**.

Code Value	Meaning
Blank	Segment is not curved
'I'	Segment is an irregular curve, i.e., it is curved but it is not an arc of a circle
'L'	Segment is an arc of a circle on the left side of the line joining the segment's FROM and TO nodes
'R'	Segment is an arc of a circle on the right side of the line joining the segment's FROM and TO nodes

DCP PREFERRED LGC

Function:	1, 1A and BL (regular WA2 only), 2, 3, 3C, BN
Work Area Format:	MSW and COW
Length and Format:	2 bytes RJZF
Description:	Identifies the local group of street names designated by the Department of City Planning as 'preferred' for display purposes for a specific location on a street.

DOGLEG FLAG

Functions:	3
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	A dogleg is a street configuration in which a street has a displacement or offset as it crosses another street. A non-blank value in the Dogleg Flag indicates that at least one of the cross streets forms a dogleg as it crosses the 'on' street, in such a way that at least one side of the 'on' street has a block face encompassing more than one segment.

When Function 3 returns a non-blank value in this flag, the work area represents the 'innermost' segment of the dogleg configuration.

The Dogleg Flag value indicates which side or sides of the street has (or have) the long block face(s).

Code	Meaning
blank	Not a dogleg
‘B’	Both sides of the ‘on’ street have long block faces formed by doglegs. This can only occur if both cross streets form doglegs as they cross the ‘on’ street.
‘L’	The left side of the street has a long block face formed by a dogleg
‘R’	The right side of the street has a long block face formed by a dogleg

A Function 3C call will return information on the long block face when the user input data specifies a side of a street where there is a long block face formed by a dogleg or doglegs. The Dogleg Flag will not be set in response to a function 3C call.

DOT STREET LIGHT CONTRACTOR AREA

Functions: 1, 1E, 2, 3, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte
Description:

Code	Meaning
‘1'	Street lights serviced by Manhattan contractors
‘2'	Street lights serviced by Bronx contractors
‘3'	Street lights serviced by Brooklyn contractors
‘4'	Street lights serviced by Queens contractors
‘5'	Street lights serviced by Staten Island contractors
‘X’	Street light is located on the Brooklyn, Queens boundary
‘N’	Street light is located in one borough, but serviced by a different borough

DYNAMIC BLOCK

Functions: 1, 1E, 3(MSW: Long WA2), 3(COW), 3C
Work Area Format: MSW and COW
Length and Format: 3 bytes RJZF
Description: A dynamic block is an un-subdivided polygon formed by LION segments. Dynamic blocks are numbered uniquely within 2000 census tract.

ELECTION DISTRICT

Function: 1E
Work Area Format: MSW and COW
Length and Format: 3 bytes
Description: A set of districts defined by the NYC Board of Elections to conduct elections. There are approximately 6,000 Election Districts (EDs) in NYC. Each ED is numbered uniquely within its Assembly District. All of NYC’s higher-level political districts, namely Assembly Districts, City Council Districts, Municipal Court Districts, Congressional Districts and State Senatorial Districts, are defined as aggregates of Eds.

EXPANDED FORMAT FLAG - See CROSS STREET NAMES FLAG

FEATURE TYPE CODE

Functions:	1, 1E, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	Identifies the type of geographic feature represented by the work area

Code	Meaning
blank	Public street that exists physically, other than a type ‘W’ feature (see below)
‘1'	Railroad
‘2'	Shoreline
‘3'	Census block boundary without physical existence
‘4'	Other non-street feature
‘5'	‘Paper street’, i.e., a public street that is legally ‘mapped’ but does not exist physically
‘6'	Private street that exists physically
‘7'	Physically nonexistent district boundary, other than a type ‘3’ feature (see above)
‘8'	Physical Boundary such as a cemetery wall
‘9'	Paper street’ that coincides with a non-physical boundary such as a Census block
‘W’	Path, non-vehicular, addressable

FIRE BATTALION

Functions:	1, 1E, 2, 3 (MSW: Long WA2), 3 (COW), 3C
Work Area Format:	MSW and COW
Length and Format:	2 bytes
Description:	An administrative fire district composed of Fire Companies.

FIRE COMPANY NUMBER

Functions:	1, 1E, 2, 3 (MSW: Long WA2), 3 (COW), 3C
Work Area Format:	MSW and COW
Length and Format:	3 bytes. Numeric RJZF
Description:	The smallest kind of administrative fire district defined by the NYC Fire Department. There are two types, indicated by the Fire Company Type: engine companies and ladder companies.

FIRE COMPANY TYPE

Functions:	1, 1E, 2, 3 (MSW: Long WA2), 3 (COW), 3C
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	Fire companies are characterized by the type of apparatus they use to fight fires.

Code	Meaning
‘E’	Engine Company
‘L’	Ladder Company

FIRE DIVISION

Functions: 1, 1E, 2, 3 (MSW: Long WA2), 3 (COW), 3C
Work Area Format: MSW and COW
Length and Format: 2 bytes
Description: An administrative fire district composed of Fire Battalions.

FUNCTION 1A/BL VERSION FLAG

Functions: 1A, BL
Work Area Format: MSW and COW
Length and Format: 1 byte
Description:

Code	Meaning
‘S’	Standard version - Required for MSW
’L’	Invalid - No Longer Supported
Blank	COW: Standard version MSW: Invalid

GAP FLAG

Functions: 3S
Work Area Format: MSW and COW
Length and Format: 1 byte
Description:

Code Value	Meaning
Blank	No gap, i.e., the ‘on’ street connects this intersection with its predecessor in list
‘G’	A gap exists along the ‘on’ street between this intersection and its predecessor

GENERATED RECORD FLAG

Functions: 3, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte
Description: This flag indicates that the geography defined by the input ‘on’ street and two cross streets is not a conventional street segment. There are several cases: a segment one of whose cross-features is a pseudo-street name (codes C, D); a street stretch formed by consolidating more than one consecutive LION segment (codes B, L, M, R, S and T); or a segment that is part of such a street stretch (types F, G). If the input data simultaneously satisfy the criteria for a Generated Record Flag value of C or D and for some other value, the flag contains the value other than C or D.

‘B’	Record has been generated by consolidating several LION segments to represent a stretch of a street where there is a node that is not at an intersection, such as a bending point (or a consecutive sequence of such nodes).
‘C’	Record generated because one or both nodes of segment lie on the City Limit (Bronx-Westchester or Queens-Nassau border), but segment itself lies entirely within the City. The cross street list for a node on the City Limit contains the special street code assigned to the pseudo-street name CITY LIMIT in the Bronx or Queens, as appropriate.
‘D’	Record has been generated for a dead end segment, i.e. a segment at least one of whose nodes either has no other segments incident at it, or has segments of non-street features only. The cross street list at such a node contains only the special street code assigned to the pseudo-street name DEAD END in the given borough.
‘F’	Record represents a segment that is part of a street stretch that either contains a bending point at which there are no cross streets, or the left side of which is the long block face of a T-intersection or a consecutive sequence of T-intersections.
‘G’	Record represents a segment that is part of a street stretch, that either contains a bending point at which there are no cross streets, or the right side of which is the long block face of a T-intersection or a consecutive sequence of T-intersections.
‘L’	Record has been generated to represent the long block face on the left side of a T-intersection.
‘M’	Record has been generated by consolidating two or more LION segments to represent a stretch of a street containing a node or a consecutive sequence of nodes at which the ‘on’ feature intersects with no streets but intersects with more than one type of non-street feature.
‘R’	Record has been generated to represent the long block face on the right side of a T-intersection.
‘S’	Record has been generated by consolidating two or more LION segments to represent a stretch of a street containing a node or a consecutive sequence of nodes at which the ‘on’ feature intersects with no streets but intersects with one or more shorelines.

'T'	Record has been generated by consolidating two or more LION segments to represent a stretch of a street containing a node or a consecutive sequence of nodes at which the 'on' feature intersects with no streets but intersects with one or more train tracks.

HEALTH AREA

Functions: 1, 1E, 2, 3, 3C
Work Area Format: MSW and COW
Length and Format: 4 bytes
Description: Districts defined by the NYC Department of Health and used to report statistics on births, deaths, communicable diseases etc. Health Areas are aggregates of Census Tracts.

HEALTH CENTER DISTRICT

Functions: 1, 1E
Work Area Format: MSW and COW
Length and Format: 2 bytes
Description: Districts defined by the NYC Department of Health for administrative purposes. Health Center Districts are aggregates of Health Areas.

HIGH HOUSE NUMBER - See HOUSE NUMBER

HOUSE NUMBER

Field Names: HOUSE NUMBER (WA1 input field, Functions 1, 1A, 1E);
NORMALIZED HOUSE NUMBER (WA1 output field, Functions 1, 1A, 1E);
LOW HOUSE NUMBER,
HIGH HOUSE NUMBER (WA2 output fields, Functions 1, 1A, 1E, 3C);
LEFT LOW HOUSE NUMBER,
LEFT HIGH HOUSE NUMBER,
RIGHT LOW HOUSE NUMBER,
RIGHT HIGH HOUSE NUMBER (WA2 output fields, Function 3)
ALTERNATE LOW HOUSE NUMBER,
ALTERNATE HIGH HOUSE NUMBER (WA2 output fields, Function 3C)
UNDERLYING HOUSE NUMBER FOR VANITY ADDRESSES (WA2, 1 and 1E-MSW: Long WA2, 1 and 1E-COW)
Functions: 1, 1A, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: See Section V.2.
Description: If the field name indicates the house number is normalized, for MSW it is in HNI format; and for COW it is in HNS format; otherwise, it is in HND format (see Section V.2).

HOUSE NUMBER JUSTIFICATION FLAG

Functions: 1, 1E, 1A, D, DG, DN
Work Area Format: COW
Length and Format: 1 byte
Description: Indicates whether the HOUSE NUMBERS IN DISPLAY format (HNDs) in the output area should be left-justified or right-justified.

Code Value	Meaning
'L' or Blank	Left-justify Normalized House Numbers (default)
'R'	Right-justify Normalized House Numbers

HOUSE NUMBER NORMALIZATION LENGTH

Functions: 1, 1E, 1A, D, DG, DN
Work Area Format: COW
Length and Format: 2 bytes, numeric
Description: Indicates the length requested for the HOUSE NUMBERS IN DISPLAY format (HNDs) in the output area. Valid values are between 12 and 16. If the field is left blank, the default is 12. For more information, see Section V.2.

INSTRUCTIONAL DIVISION - See INSTRUCTIONAL REGION

INSTRUCTIONAL REGION

Functions: 1, 1E, 2, 3, 3C
Work Area Format: MSW and COW
Length and Format: 2 bytes
Description: The Department of Education has divided the city into Instructional Regions which group together two or more Community School Districts for administrative purposes. (Instructional Regions have also been known as Instructional Divisions.)

INTERIM ASSISTANCE ELIGIBILITY INDICATOR

Functions: 1, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte
Description: Indicates whether the input location is in a census tract that is Community Development eligible.

Code Value	Meaning
'E'	Input location is in a CD-eligible census tract
'I'	Location is not in a CD-eligible census tract
Blank	Location is in a census tract, the CD-eligibility status of which is unknown to the Geosupport System. (Note: This is an error condition and should be reported).

INTERSECTION REPLICATION COUNTER

Functions: 2
Work Area Format: MSW and COW
Length and Format: 1 byte, numeric
Description: The Intersection Replication Counter is non-blank only if the two streets intersect more than once, in which case this field contains the number of such intersections.

INTERIOR LOT FLAG

Functions:	1A, BL, BN	
Work Area Format:	MSW and COW	
Length and Format:	1 byte	
Description:	Code Value	Meaning
	'I'	Tax lot is interior to physical block, ie., it has no street frontages.
	Blank	Tax lot has at least one street frontage

INTERNAL LABEL POINT - See SPATIAL COORDINATES OF THE INTERNAL LABEL POINT OF THE TAX LOT

IRREGULARLY-SHAPED LOT FLAG

Functions:	1A, BL, BN	
Work Area Format:	MSW and COW	
Length and Format:	1 byte	
Description:	Code Value	Meaning
	'I'	Tax lot is irregularly-shaped, i.e., non-rectangular
	Blank	Tax lot is rectangular

LENGTH IN FEET FROM PREVIOUS NODE

Functions: 3S
Work Area Format: MSW and COW
Length and Format: MSW: 3 bytes packed; COW: 5 bytes numeric RJZF
Description: The length between two nodes.

LGI OVERFLOW FLAG

Functions: 1A and BL - regular WA2, BN
Work Area Format: MSW and COW
Length and Format: 1 byte
Description: If set to 'E' indicates that the number of geographic identifiers for the given tax lot exceeds 21, the maximum capacity of the List of Geographic Identifiers; otherwise it is blank. If this flag is set to 'E', the user can obtain a comprehensive list of BINs for the tax lot by using the long Work Area 2 option when calling the same function with the same input data.

LION FACE CODE

Functions: 1, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: 4 bytes. Numeric
Description: A LION Face Code is assigned to each linear geographic feature represented in the LION file. These consist of streets and certain non-street features, such as census boundaries, shorelines and railroad tracks. Face Codes serve as part of LION keys, which identify a unique LION record. Face Code values are assigned uniquely within borough.

LION NODE NUMBER

Functions: 2, 3S (COW Only)
Work Area Format: MSW and COW
Length and Format: 7 bytes. Numeric
Description: A LION node is an endpoint of a geographic feature segment represented in LION. Most nodes in LION are points where a feature bends or terminates or where two features intersect. Each LION node has a node number assigned to it, which is unique in the entire city. Node number assignments are permanent; if a node is deleted from LION, its node number is retired and is never reassigned to a different node.

LION SEGMENT-ID

Functions: 1 and 1E (MSW: Long WA2 only), 1 and 1E (COW), 3, 3C
Work Area Format: MSW and COW
Length and Format: 7 bytes. RJZF
Description: Identifies, uniquely within the entire city, a geographic feature segment represented in the LION file.

LION SEQUENCE NUMBER

Functions: 1, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: 5 bytes
Description: Identifies a LION record uniquely within Face Code. Generally, Sequence Numbers are assigned in the geographic order in which the corresponding LION segments occur along the geographic feature identified by the given face code. The Borough Code, Face Code and Sequence Number concatenated form the LION key, which serves as a unique identifier for one LION record.

LIST OF BUILDINGS ON TAX LOT

Functions: 1A and BL - Long WA2 only
Work Area Format: MSW and COW
Length and Format: 17,500 bytes, consisting of 2,500 slots for 7-byte BINs
Description: See Section VI.6.

LIST OF CROSS STREET CODES

Functions: 3S
Work Area Format: MSW and COW
Length and Format: MSW: 8 bytes, packed decimal, consisting of 2 slots for intersecting PB5SCs.
COW: 40 bytes, numeric, consisting of 5 slots for intersecting B7SCs.
Description: MSW: For each intersecting street, this is the lowest and second lowest PB5SCs for the cross streets.
COW: For each intersecting street, this is a list of up to five B7SCs, starting with the lowest B7SC, followed by the next lowest, followed by the remaining B7SCs in any order.

LIST OF CROSS STREETS

Field Names:	LIST OF CROSS STREETS AT LOW ADDRESS END, LIST OF CROSS STREETS AT HIGH ADDRESS END
Functions:	1, 1E, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	MSW: 20 bytes, consisting of slots for up to five 4-byte PB5SCs. ‘Empty’ slots contain packed zeros. COW: 30 bytes, consisting of slots for up to five 6-byte B5SCs. ‘Empty’ slots contain either numeric zeros or blanks.
Description:	A list of PB5SCs for MSW and B5SCs for COW, for up to five streets incident upon a delimiting node (endpoint) of a block face or street segment. The number of non-empty list entries is returned in the corresponding WA2 field NUMBER OF CROSS STREETS AT (LOW or HIGH) ADDRESS END. It is possible for the list to be entirely empty. If the node lies on a borough boundary, the list may contain streets from both boroughs. Subject to the space limitation, the list may include the input cross street corresponding to the given node, and may include the pseudo-streets ‘City Limit’, ‘Dead End’ and ‘Bend’. The inclusion of ‘Bend’ in the list indicates that the node is a bending point of the ‘on’ street, not that it is a bending point of a cross street (although that may be true).

LIST OF CROSS STREETS AT HIGH ADDRESS END - See LIST OF CROSS STREETS

LIST OF CROSS STREETS AT LOW ADDRESS END - See LIST OF CROSS STREETS

LIST OF GEOGRAPHIC IDENTIFIERS

Functions:	1A and BL - regular WA2, BN
Work Area Format:	MSW and COW
Length and Format:	756 bytes total, consisting of space for 21 36-byte entries, each entry having fields for the following data items: Low House Number of Address Range High House Number of Address Range B5SC DCP-Preferred LGC BIN Entry Type Code Side of Street Indicator.
Description:	The List of Geographic Identifiers (LGI) is intended to provide a comprehensive geographic profile of a tax lot by listing, so far as the information is known and space allows, all of the lot’s buildings; all of the street addresses and non-addressable street frontages of each building; all of the lot’s ‘vacant frontages’ (i.e., street frontages of the lot not associated with buildings); and any NAPs associated with the lot. The LGI contains space for up to 21 entries. The number of non-empty entries is indicated in the WA2 field NUMBER OF GEOGRAPHIC IDENTIFIERS. The types of entries that the LGI can contain are as follows:

<u>List of Geographic Identifiers - Possible Entry Types</u>

Entry Type Code	Represents	Description
Blank	Address range	A real address range of a building on a given tax lot. There are values in the Low House Number, High House Number, B5SC, DCP-Preferred LGC, Side of Street Indicator and BIN fields. A single address is represented as an address range in which the low and high house numbers are identical.
B	NAUB	A Non-Addressable Un-named Building (NAUB) (see Section VI.3). The Low and High House Number and Side of Street Indicator fields are blank. The B5SC and DCP-Preferred LGC fields usually contain the street code and LGC, correspondingly, of the street nearest to or most accessible to the NAUB, but they may be blank. The BIN field contains a meaningful value. Note: If the NAUB has frontages on more than one street, there are multiple type B entries to represent all of the NAUB's street frontages.
F	Vacant Street Frontage	A street frontage of the tax lot at which there are no buildings (including NAUBs) and to which no pseudo-addresses have been assigned. The Low and High House Number, BIN and Side of Street Indicator fields are empty. There are values in the B5SC and DCP-Preferred LGC fields.
G	NAP of a Complex	A Non-Addressable Place name (NAP) of a complex of buildings and/or other geographic features, usually on a large site or superblock (see Section III.6). The house number and BIN fields are empty. The B5SC, DCP-Preferred LGC, and Side of Street Indicator fields contain the values of these items assigned to the NAP.
N	NAP of a Simplex	A NAP of a building or other geographic feature that is not part of a complex (see Section III.6). The house number fields are empty. The B5SC, DCP-Preferred LGC, and Side of Street Indicator fields contain the values of these items assigned to the given NAP. The BIN field is non-empty only if the NAP represents a building.
Q	Pseudo-Address Range	A pseudo-address range assigned to a vacant street frontage of the tax lot. There are values in the Low House Number, High House Number, B5SC, DCP-Preferred LGC and Side of Street Indicator fields. A single address is represented as an address range in which the low and high house numbers are identical. The BIN field is empty.
R	Real Street of a Vanity Address	Entry indicates the street and the side of that street on which the building entrance having a vanity address is really located and for which no other address for that building exists. For a discussion of vanity addresses, see Section V.9. In a type R entry, the Low and High House Number fields are empty, and there are non-empty values in the B5SC, DCP-Preferred LGC, Side of Street Indicator

		and BIN fields. Whenever the LGI contains a type R entry, it also contains a type V entry for the associated vanity address.
V	Vanity Address	A vanity address or address range. For a detailed discussion of vanity addresses, see Section V.9. There are non-empty values in the Low House Number, High House Number, B5SC, DCP-Preferred LGC, Side of Street Indicator and BIN fields. A single address is represented as an address range in which the low and high house numbers are identical. Whenever the LGI contains a type V entry, it also contains an either an address range entry or a type R entry that indicates the street on which the associated building entrance is really located.
W	Blank-Wall Bldg Frontage	A building frontage along a street that is not associated with any addresses, such as some building facades with no entrances. The Low and High House Number and Side of Street Indicator fields are blank. There are values in the B5SC and DCP-Preferred LGC fields. The BIN field contains a meaningful value. Note: Type W entries exist only for buildings that also have at least one real address range entry. If a building has no real address ranges, the building is a NAUB, and its street frontages, if any, are represented by type B entries rather than type W entries.
X	NAP of a Constituent Entity of a Complex	A NAP of a constituent entity of a complex. (The NAP of the entire complex is represented by a separate entry of type G.) The house number fields are empty. The B5SC, DCP-Preferred LGC and Side of Street Indicator fields contain the values of these items assigned to the NAP. The BIN field is non-empty only if the NAP represents a building.

The combination of fields in an LGI entry that contain information depends on the entry type, as indicated in the following table:

List of Geographic Identifiers - Which Fields Contain Values By Entry Type

Entry Type Code	Entry Type	Low & High House Numbers	B5SC	LGC	Side of Street Indicator	BIN
blank	Real Address Range	✓	✓	✓	✓	✓
B	NAUB		(*)	(*)		✓
F	Vacant Street Frontage		✓	✓		
G	NAP of a complex		✓	✓	✓	
N	NAP of a simplex		✓	✓	✓	(**)
Q	Pseudo-Address Range	✓	✓	✓	✓	
R	Real Street of Vanity Address		✓	✓	✓	✓
V	Vanity Address	✓	✓	✓	✓	✓
W	Blank-Wall Building Facade		✓	✓		✓
X	Nap of a constituent entity of a complex		✓	✓	✓	(**)

(*) NAUB entries may or may not contain B5SC and LGC values. An entry for a NAUB contains a B5SC value and an LGC value only if the GSS staff has determined that the NAUB fronts on, is adjacent to or is principally accessible from a particular street.

(**) The BIN field is non-empty only if the NAP represents a building.

The LGI's entries are ordered so that entries with non-empty BINs are listed first, grouped by BIN. Except for a special case (alternative borough for Marble Hill and Rikers Island - see Section V.7), if the input address is a real address, the first group of entries in the LGI are those for the BIN corresponding to the input address, and (except for alternative street records for Ruby Street - see Section V.8) the address range encompassing the input address is the very first entry in the LGI. After all the entries with non-empty BINs are listed, any entries with empty BINs, such as entries for pseudo-address ranges, are listed in no particular order as space allows. If the input address is a pseudo-address range, it may or may not appear in the LGI, depending on space and on the order in which the non-BIN entries happen to be listed.

LIST OF GEOGRAPHIC IDENTIFIERS OVERFLOW FLAG - See LGI OVERFLOW FLAG

LIST OF INTERSECTING STREETS

Function:	2
Work Area Format:	MSW and COW
Length and Format:	MSW: 20 bytes, consisting of slots for up to five 4-byte PB5SCs. 'Empty' slots contain packed zeros. COW: 30 bytes, consisting of slots for up to five 6-byte B5SCs. 'Empty' slots contain numeric zeros or blanks.

Description:	A list of PB5SCS for MSWs and B5SCs for COWS for up to five streets incident upon the intersection. The field NUMBER OF INTERSECTING STREETS contains the number of non-empty entries in the list. If the intersection lies on a borough boundary, the list may contain streets from both boroughs. Subject to the space limitation, the list may include the two input streets, and may include the pseudo-streets 'City Limit' and 'Dead End'. The list never includes the pseudo-street 'Bend'. The list always contains at least one entry (it contains precisely one entry in the case of a bending point of a street at which there are no other streets).

LIST OF STREET CODES

Functions:	1*, 2, 3*, BB, BF
Work Area Format:	COW
Length and Format:	80 bytes, consisting of 10 fields for B7SCs
Description:	List of borough and 7-byte street codes, corresponding to the LIST OF STREET NAMES. The number of street codes in the list is returned in the WA1 output field NUMBER OF STREET CODES AND STREET NAMES IN LIST.

LIST OF STREET NAMES (WA1 output field)

Functions:	1*, 2, 3*, BB, BF
Work Area Format:	MSW and COW
Length and Format:	320 bytes, consisting of 10 fields for street names, each 32 bytes.
Description:	This field is used by several Geosupport features (see below) to return a list of street names. The number of street names in the list is returned in the WA1 output field NUMBER OF STREET NAMES IN LIST for MSWs, and in the WA1 output field NUMBER OF STREET CODES AND STREET NAMES IN LIST for COWs.

The similar names feature uses the List of Street Names to return up to ten street names deemed 'similar' to a rejected input street name (see Section III.5).

The browse functions, Functions BB and BF, use the List of Street Names to return up to ten normalized street names in alphabetical order as part of a street name browse (see Section III.7).

The local street name validation feature uses the List of Street Names to return up to four locally valid alias street names corresponding to a street name rejected as locally invalid (see Section IV.5).

The cross street names feature (see CROSS STREET NAMES FLAG) uses the List of Street Names to return street names corresponding to the street codes in the LIST OF CROSS STREETS (Functions 1, 1E, 3 and 3C) or the LIST OF INTERSECTING STREETS (Function 2). In the case of Functions 1, 1E, 3 and 3C, the first five 32-byte street name fields in the List of Street Names are used for the street names corresponding to the street codes in the LIST OF CROSS STREETS AT LOW ADDRESS END; the second five 32-byte street name fields in the List of Street Names are used for the street names corresponding to the street codes in the LIST OF CROSS STREETS AT HIGH ADDRESS END.

LOCATIONAL STATUS OF SEGMENT

Functions: 3, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte
Description: Indicates locational status of segment per codes below.

Code	Meaning
‘H’	Segment internal to a block, but not a Dead End (A Land Hook)
‘I’	Dead End Segment
‘X’	Tract Boundary Segment (other than borough boundary)
‘1'	Segment bordering Manhattan
‘2'	Segment bordering Bronx
‘3'	Segment bordering Brooklyn
‘4'	Segment bordering Queens
‘5'	Segment bordering Staten Island
‘9'	Segment bordering City Limits

LOW HOUSE NUMBER - See HOUSE NUMBER

NORMALIZED HOUSE NUMBER - See HOUSE NUMBER

NUMBER OF BUILDINGS ON TAX LOT

Functions: 1A, BL - long WA2 only
Work Area Format: MSW and COW
Length and Format: 4 bytes numeric.
Description: Indicates the number of entries in the LIST OF BUILDINGS. Maximum value is 2,500.

NUMBER OF CROSS STREETS

Field Names: NUMBER OF CROSS STREETS AT LOW ADDRESS END, NUMBER OF CROSS STREETS AT HIGH ADDRESS END
Functions: 1, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: One byte, containing a numeric digit from 0 to 5.
Description: Indicates the number of non-empty entries in the corresponding LIST OF CROSS STREETS.

NUMBER OF CROSS STREETS AT HIGH ADDRESS END - See NUMBER OF CROSS STREETS

NUMBER OF CROSS STREETS AT LOW ADDRESS END - See NUMBER OF CROSS STREETS

NUMBER OF GEOGRAPHIC IDENTIFIERS

Functions:	1A and BL - regular WA2 only, BN
Work Area Format:	MSW and COW
Length and Format:	2 bytes for MSW; 4 bytes for COW
Description:	Indicates the number of entries in the LIST OF GEOGRAPHIC IDENTIFIERS.

NUMBER OF INTERSECTING STREETS

Functions:	2
Work Area Format:	MSW and COW
Length and Format:	One byte, containing a numeric digit from 1 to 5.
Description:	Indicates the number of non-empty entries in the LIST OF INTERSECTING STREETS.

NUMBER OF STREET FRONTAGES OF LOT

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	2 bytes, RJZF.
Description:	Indicates the number of streets on which the given lot has at least one frontage.

NUMBER OF STREET CODES AND STREET NAMES IN LIST

Functions:	1, 2, 3*, BB, BF
Work Area Format:	COW
Length and Format:	2 btyes, Numeric
Description:	Indicates the number of street names returned in the LIST OF STREET NAMES, corresponding to the number of street codes returned in the LIST OF STREET CODES..

NUMBER OF STREET NAMES IN LIST (WA1 output item)

Functions:	1, 2, 3*, BB, BF
Work Area Format:	MSW
Length and Format:	2 btyes, packed decimal
Description:	Indicates the number of street names returned in the LIST OF STREET NAMES.

PLATFORM INDICATOR - See WORK AREA FORMAT INDICATOR

POLICE PATROL BOROUGH COMMAND

Functions:	1, 1E, 2, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	These are sub-borough geographic areas defined by the Police Department. They are composed of Police Precincts.

Code	Meaning
1	Manhattan South
2	Manhattan North
3	Bronx
4	Brooklyn South
5	Brooklyn North
6	Queens North
7	Staten Island
8	Queens South

POLICE PRECINCT

Functions:	1, 1E, 2, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	3 bytes. RJZF.
Description:	Police Patrol Borough Commands are sub-divided into Police Precincts which are defined by the Police Department.

RPAD BUILDING CLASSIFICATION CODE

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	2 bytes
Description:	This is a set of land use/building classification codes defined by the Real Property Assessment Division (RPAD) of the Department of Finance. If a tax lot has more than one building or land use, RPAD assigns the building class code they deem to describe best the ‘principal’ building or the ‘predominant’ land use on the tax lot. The values and meanings of this set of codes can be obtained from the Department of Finance.

RPAD CONDO IDENTIFICATION NUMBER

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	4 bytes
Description:	This is an identification number assigned by the Department of Finance to each condominium in the city. It identifies the condominium as a whole and not a specific condominium unit.

RPAD SELF-CHECK CODE (SCC) FOR BBL

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	For each BBL value, the Department of Finance has computed a Self-Check Code (SCC). This is a one-digit number computed from the BBL value using a algorithm

chosen by DOF. The purpose of the SCC is to assist in validating key-entered BBLs. For more information on SCCs inquire to the information technology division of the Department of Finance.

ROADBED REQUEST SWITCH

Functions:	1, 1E
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	Indicated request for Roadbed information for roads that are divided into two or more roadbeds. If Roadbed information if requested for a street that is not divided, Geosupport returns the generic information. The Segment Type Code will indicate the type of information that is being returned.

Code	Meaning
'R'	Roadbed information requested
Blank	Generic (non-roadbed) information requested (default)

SANBORN VOLUME AND PAGE

Functions:	1A, BL, BN, 2
Work Area Format:	MSW and COW
Length and Format:	The Volume field is 3 bytes (2-digit volume number + 1-digit character suffix). The Page field is 4 bytes (3-digit page number + 1-digit character suffix).
Description:	The Sanborn Map Company maintains a 79 volume atlas of New York City geography that is widely used by New York city agencies. The atlases contain approximately 6000 maps covering all five boroughs.

SANITATION COLLECTION SCHEDULING SECTION AND SUBSECTION

Functions:	1, 1E
Work Area Format:	MSW and COW
Length and Format:	2 bytes
Description:	Districts defined by the Department of Sanitation for waste collection.

SANITATION RECYCLING PICKUP

Functions:	1, 1E
Work Area Format:	MSW and COW
Length and Format:	3 bytes
Description:	Indicates which days of the week the Department of Sanitation will pick up recycling at the given address.

SANITATION REGULAR PICKUP

Functions:	1, 1E
Work Area Format:	MSW and COW
Length and Format:	3 bytes
Description:	Indicates which days of the week the Department of Sanitation will pick up non-recycling waste at the given address.

SEGMENT AZIMUTH

Functions: 3, 3C
Work Area Format: MSW and COW
Length and Format: 3 bytes
Description: This item represents the direction in which the segment lies on the earth's surface, expressed as an angle in degrees measured counterclockwise from due east. The segment is considered to be pointing in the direction of increasing addresses, and the azimuth value can range from 0 to 359 degrees, inclusive. For example, a segment pointing due east has an azimuth of 0; one pointing due north has an azimuth of 90; one pointing due west has an azimuth of 180; one pointing halfway between due west and due south (i.e, pointing due southwest) has an azimuth of 225.

SEGMENT ID - See LION SEGMENT ID

SEGMENT LENGTH IN FEET

Functions: 1, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: MSW: Fns 1 and 1E **-** 5 bytes numeric, Fns 3 and 3C **-** 3 bytes packed decimal
COW: Fns 1, 1E, 3, 3C **-** 5 bytes numeric
Description: Except for curved segments (**see Curve Flag**), the Segment Length is computed from the **Spatial Coordinates** of the segment's endpoints, as digitized in the LION file. For curved segments, the Segment Length is computed by summing the lengths of the small straight line segments that approximate the curve in the GIS version of LION; this is a more accurate approximation to the true arc length of the curve than would be the segment's 'secant length', that is, the straight line distance between the curve's extreme endpoints. In the case of Functions 3 and 3C, if the input data define a street stretch encompassing more than one segment (because of a T-intersection or bend), the Segment Length returned is the sum of the lengths of the constituent segments of the stretch. In all cases, the Segment Length has a very approximate level of accuracy only, and should not be used in applications requiring high precision.

SEGMENT ORIENTATION

Functions: 3, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte character
Description: This item is a set of codes grouping the possible azimuth values of a segment into eight categories. The categories are "approximately" due north, south, east and west, and the four quadrants of the rectangular coordinate system for segments that do not lie approximately due north, south, east or west. "Approximately" as used here means "within 5 degrees". In Manhattan, all orientation codes are defined with a 30-degree clockwise shift (i.e., 30 is subtracted from the azimuth value) in order to conform to the conventional concept that the midtown streets and avenues lie due east-west and due north-south, respectively. For example, "approximately due north" means "within 5 degrees of due north"; for the boroughs other than Manhattan, this corresponds to the range of azimuth values from 85 to 95; in Manhattan, the corresponding azimuth value range is 55 to 65. There is a ninth orientation category, with a code value of 'U', meaning Geosupport could not

determine the segment's orientation because of a problem with the segment's **Spatial Coordinates**. All occurrences of an orientation code of 'U' should be reported to Geographic Systems Section staff.

Code Value	Meaning	Corresponding Range of Azimuth Values: Manhattan	Other Boroughs
U	Orientation is undefined		
E	Approximately due east	0-5 and 355-359	325-335
1	First quadrant, i.e. northeasterly	6-84	336-359 and 0-54
N	Approximately due north	85-95	55-65
2	Second quadrant, i.e. northwesterly	96-174	66-144
W	Approximately due west	175-185	145-155
3	Third quadrant, i.e. southwesterly	186-264	156-234
S	Approximately due south	265-275	235-245
4	Fourth quadrant, i.e. southeasterly	276-354	246-324

SEGMENT TYPE CODE

Functions: 1, 1E, 3, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte
Description: Indicates type of segment.

Code Value	Meaning
'U'	Undivided
'G'	Generic
'B'	Both Generic and Roadbed
'R'	Roadbed
'C'	Connector
'E'	Exit/Entrance Ramp
'T'	Terminator
'F'	Faux Ramps (ramp segments connecting the roadbed segments to the generic segments)

SIDE OF STREET INDICATOR

Functions: 1, 1E, 1A - regular WA2, 3C
Work Area Format: MSW and COW
Length and Format: 1 byte character
Description: This field indicates on which side of the street, left or right, the block face containing the input address lies. Left and right are defined with respect to the direction of <u>increasing</u> addresses along the 'on' street.

Code Value	Meaning
L	Block face is on left side of street with respect to direction of increasing address
R	Block face is on right with respect to direction of increasing address

SPATIAL COORDINATES

Functions: 1, 1E, 2

Work Area Format: MSW and COW

Length and Format: Spatial coordinates consist of two fields, an X Coordinate and a Y Coordinate, each 7 bytes RJZF.

Description: Spatial coordinates are a pair of numbers that specify a location on the earth's surface. Geosupport returns spatial coordinates for an input address (Functions 1 and 1E) or intersection (Function 2). Spatial coordinates are often used in conjunction with separate computer mapping and Geographic Information System (GIS) software to generate maps and for spatial analysis, although the Geosupport System does not itself provide users with such capabilities. **Note: For Functions 1 and 1E, the spatial coordinates that Geosupport returns are imprecise approximations of real-world locations, and are not appropriate for use in applications that require a high level of spatial accuracy**.

Spatial coordinates are expressed various geodetic coordinate systems, of which latitude/longitude is a well-known example. The coordinate system that Geosupport uses is known as the State Plane Coordinate (SPC) system. The SPC system is based upon the fact that, in a small enough geographic area, the earth's surface can be assumed to be flat without introducing a significant error. In the SPC system, each state of the U.S. is subdivided into zones small enough to model as planar areas. In each SPC zone, a Cartesian coordinate system is established, with the X and Y coordinate axes oriented due east and due north, respectively, and the origin selected to be a point well to the southwest of the entire zone. (The origin is so selected to insure that the X and Y coordinates of all points within the zone are positive values.) The SPC zone that New York City is in, and which Geosupport uses, is called the New York-Long Island zone, NAD 83. In the SPC system, one unit of X or Y represents one foot of distance on the ground. A major advantage of the SPC system over other map projection systems is the ease of calculating the distance between two points.

In the case of Functions 1 and 1E, if the street segment on which the input address lies is a straight line segment or an arc of a circle, Geosupport computes and returns output spatial coordinates using a complex algorithm, a detailed description of which is beyond the scope of this document. If, however, the input address lies on a irregularly curved geographic feature (see **Curve Flag**), Functions 1 and 1E return blanks in the spatial coordinate fields.

Function 1/1E's spatial coordinates algorithm produces a point position based on how the input address is prorated with respect to the administrative address range allocated to the entire block face. In addition, the computed point is positioned slightly set off from the segment, on the side of the street where the input address is located. This offset is graphically desirable and also insures that the point will fall within the interiors of the proper political and administrative district boundary polygons for the given address. The computed point is a rough approximation to the location of the input address, intended to be used only for thematic mapping and other purposes that do not require a high level of spatial accuracy. If the input address is a vanity address (see Section V.9), the computed point is based on the street segment where the address is actually located. In general, points computed for vanity addresses have even less spatial accuracy than points computed for conventional addresses.

In the case of Function 2, the spatial coordinates returned are those of the LION node that corresponds to the input street intersection. Those coordinates represent an approximate center point of the intersection.

SPATIAL COORDINATES OF THE INTERNAL LABEL POINT OF THE TAX LOT

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	Spatial coordinates consist of two fields, an X Coordinate and a Y Coordinate, each 7 bytes RJZF.
Description:	The Internal Label Point is a location within a tax lot selected by the Department of City Planning as the location where information could be displayed about the property in a mapping application. The coordinates associated with the Internal Label Point are guaranteed to be within the property, unlike the coordinates returned by either Function 1 or Function 1E, where the Spatial Coordinates are an approximation based in the address range of the particular street the address is on. In addition, the Function 1/1E Spatial Coordinates always fall in the street bed and not within a tax lot, and most likely will not be adjacent to the tax lot the address is in. Additionally, when using Function 1A, the same coordinates will be returned no matter which of a tax lot's addresses is used as input. There are a few properties which do not have an Internal Label Point; consequently, no coordinates will be returned for these properties.

Internal Label Points have also been known as Annotation Points.

See SPATIAL COORDINATES for a description of the coordinate system (SPC) used by Geosupport.

SPECIAL ADDRESS GENERATED RECORD FLAG

Functions:	1, 1E
Work Area Format:	MSW and COW
Length and Format:	1 byte character
Description:	A non-blank value in this flag indicates one of a variety of special addressing situations.

Code Value	Meaning
‘A’	The address range returned in this work area is alternative to the address range that is stored in LION for this block face. This case arises most commonly when the input address is an old (superseded) address on a block face on which the buildings were re-numbered at some time in the past. For such an input address, the address range returned in this work area is the old address range, whereas the current address range is stored in LION. Another situation in which this flag is ‘A’ is when the given street segment has continuous parity address ranges on both sides of the street, such as when buildings are numbered consecutively around the arc of a cul-de-sac.
‘B’	The input street name or five-digit street code is different from that stored in LION for this block face. This case arises when two street names having different B5SCs are both valid along a street or portion of a street. These are situations in which treating the two street names as aliases would result in an address range overlap.
‘C’	The input address pertains to Ruby Street, a street along the Brooklyn-Queens border that has a unique addressing situation. See SectionV.8.
‘D’	The input address involves a duplicate address situation. See Section V.6.
‘E’	The input address is in one of the neighborhoods in which the name of the neighborhood can serve as an alternative street name for the streets in that neighborhood. Two Bronx neighborhoods, Edgewater Park and Harding Park, have this characteristic.
‘G’	The input name or street code corresponds to a non-addressable place name of a complex. A complex is a geographic feature that contains constituent entities that are separately geographically identifiable. Typical examples of complexes include airports, housing projects and university and hospital campuses. See Section III.6.
‘N’	The input name or street code corresponds to a non-addressable place name of a “stand-alone” geographic feature (a geographic feature that is neither a complex nor a constituent entity of a complex). Typical examples are individual named buildings, such as Empire State Building, Shea Stadium, Carnegie Hall. See Section III.6.
‘O’	The block face contains out-of-sequence and/or opposite-parity addresses. An out-of-sequence address contains a house number that is out of sequence with those of the immediately adjacent buildings. An opposite-parity address contains a house number that is of the opposite parity to the predominant parity on the block face. See Section V.10.
‘P’	The input address contains an addressable place name. Example: 2 Penn Plaza. See Section III.6.
‘S’	The input address contains a house number suffix and is either the first or last address on this block face.

‘V’	The input address is a ‘vanity address’, that is, an address in which the street name refers to a different street than the one on which the referenced building entrance is actually located. See Section V.9.
‘X’	The input data specify a non-addressable place name of a constituent entity of a complex. Examples: AVERY FISHER HALL and NEW YORK STATE THEATER are names of constituent entities of the complex LINCOLN CENTER. See Section III.6.

SPLIT COMMUNITY SCHOOL DISTRICT FLAG

Functions: 1, 1E
Work Area Format: MSW and COW
Length and Format: 1 byte.
Description:

Code Value	Meaning
‘S’	Block face is split among two or more school districts
blank	Block face lies entirely within a single school district

SPLIT ELECTION DISTRICT FLAG

Function: 1E
Work Area Format: MSW and COW
Length and Format: 1 byte.
Description:

Code Value	Meaning
‘S’	Block face is split among two or more election districts
blank	Block face lies entirely within an election district

STATE SENATORIAL DISTRICT

Function: 1E
Work Area Format: MSW and COW
Length and Format: 2 bytes. Numeric.
Description: A district of the upper house of the New York State legislature. Consists of an aggregation of Election Districts

STREET ATTRIBUTE INDICATOR

Functions: WA1 output field - 1, 1A, 1E, 1N D, DG, DN
Work Area Format: MSW and COW
Length and Format: 1 byte character
Description: Indicates certain characteristics of selected streets.

Code Value	Meaning
‘E’	Input street is entirely in Edgewater Park (a Bronx neighborhood that has special addressing characteristics).
‘F’	Input street is partly in Edgewater Park and partly outside of it.

‘G’	NAP of a complex.
‘H’	All house numbers on input street are hyphenated.
‘I’	Named Intersection
‘M’	House numbers on input street are of mixed hyphenation, i.e., some are hyphenated and some are not.
‘N’	Input street is a Non-Addressable Place Name (NAP).
‘S’	Front Truncated Street Name.
‘X’	Nap Of a Constituent Entity of a Complex
Blank	None of the above. In particular, all addresses on the input street are un-hyphenated.

STREET NAME NORMALIZATION FORMAT FLAG

Functions: 1, 1A, 1E, 1N, 2, 3, 3C, 3S, D, DG, DN
Work Area Format: MSW and COW
Length and Format: 1 byte.
Description: Specifies the format in which Geosupport is to return output normalized street names. The default is to return street names in the sort format.

Code Value	Meaning
blank	Return normalized street names in the sort format
‘C’	Return normalized street names in the compact format
‘S’	Return normalized street names in the sort format

STREET NAME NORMALIZATION LENGTH LIMIT (SNL)

Functions: WA1 input item, Functions 1, 1A, 1E, 1N, 2, 3, 3C, 3S, D, DG, DN
Work Area Format: MSW and COW
Length and Format: 2 bytes, blank or numeric, either LJBF or RJZF.
Description: Specifies the maximum length in bytes within which Geosupport is to normalize street names. The minimum and maximum permissible SNL values are 4 and 32. The default that is in effect if the application does not specify an SNL value is 32.

STROLLING KEY - *not implemented*

Functions: 1A, BL
Work Area Format:
Length and Format:
Description:

TAX BLOCK

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	5 bytes
Description:	See Section VI.2.

TAX LOT

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	4 bytes
Description:	See Section VI.2.

TAX LOT VERSION NUMBER - *not implemented*

Functions:	
Work Area Format:	
Length and Format:	
Description:	

TAX MAP PAGE - *not implemented*

Functions:	1A, BL, BN
Work Area Format:	
Length and Format:	2 bytes
Description:	See description at **Tax Map Section.** Tax Map Page values are unique within Tax Map Section and Volume.

TAX MAP SECTION

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	2 bytes
Description:	The Department of Finance real property tax maps are organized into sections; each section is organized into volumes; and each volume consists of pages. Tax Map Section values are unique within borough.

TAX MAP VOLUME

Functions:	1A, BL, BN
Work Area Format:	MSW and COW
Length and Format:	2 bytes
Description:	See description at **Tax Map Section.** Tax Map Volume values are unique within Tax Map Section.

VACANT LOT FLAG

Functions:	1A, BL	
Work Area Format:	MSW and COW	
Length and Format:	1 byte.	
Description:	<u>Code Value</u>	<u>Meaning</u>
	'V'	Tax lot is currently vacant, i.e., it has no existing buildings
	Blank	Tax lot has at least one existing building

WORK AREA FORMAT INDICATOR

Functions:	All
Work Area Format:	MSW and COW
Length and Format:	1 byte
Description:	This indicator specifies which work area layouts are to be used in an API call. Note: This indicator is also known as the Platform Indicator.

Code	Meaning
blank	The IBM mainframe specific work areas (MSWs) are used. The MSWs contain packed decimal fields. In general, these work areas are the ones described throughout this manual.
'C'	The platform-independent work areas known as the Character-Only Work Areas (COWs) are used. These contain no packed decimal fields. For information on using COWs on the mainframe and the differences from the MSWs, see Appendix 12. For the work area layouts of the COWs, see Appendix 13.

XY COORDINATES - see SPATIAL COORDINATES

ZIP CODE

Functions:	1, 1E, 2, 3, 3C
Work Area Format:	MSW and COW
Length and Format:	5 bytes. Numeric.
Description:	U.S. Postal Service's 5-digit zip code.

APPENDIX 4: GEOSUPPORT RETURN CODES, REASON CODES AND MESSAGES

This appendix consists of a table listing all of the Geosupport Return Codes (GRCs), Reason Codes and Messages, and the Geosupport function(s) that can elicit each of them. The table is current as of the Geosupport software version indicated in the table heading. See Section II.2 for a detailed discussion of application reject handling using GRCs, Reason Codes and Messages.

In the table, an asterisk in the second byte position of a function code is a ‘wildcard’, signifying all function codes that begin with the character that is in the first byte position. For example, ‘3*’ signifies all function codes that begin with ‘3’ (namely, as of this writing, Functions 3, 3C and 3S).

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
00		All	[Processing was unconditionally successful-no message issued]
01	**[GRC 01s are warnings]**		
	1	1,1A,1E	ADDR NUMBER ALTERED: RANGE ASSUMED.USING DIGITS BEFORE DASH ONLY
	2	1,1A,1E	ADDR NUMBER ALTERED: HYPHEN INSERTED
	3	1,1A,1E	ADDR NUMBER ALTERED: HYPHEN DELETED
	4	BB,BF	YOU HAVE REACHED THE <FIRST or LAST> STREET NAME IN THE BOROUGH OF <boro. name>
	5	1,1A,1E	INPUT IS A COMPLEX. OUTPUT DATA MAY PERTAIN TO ONLY PART OF THE COMPLEX
	6	1,1A,1E	OUTPUT STREET NAME/CODE DIFFER FROM INPUT
	7	1,1A,1E	OUTPUT STREET NAME/CODE DIFFER FROM INPUT. ADDR NUMBER ALTERED: RANGE ASSUMED OUTPUT STREET NAME/CODE DIFFER FROM INPUT. ADDR NUMBER ALTERED: HYPHEN INSERTED OUTPUT STREET NAME/CODE DIFFER FROM INPUT. ADDR NUMBER ALTERED: HYPHEN DELETED
	8	1A	INPUT ADDRESS IS A PSEUDO-ADDRESS
	9	1A	INPUT ADDRESS IS A PSEUDO-ADDRESS. ADDR NUMBER ALTERED: RANGE ASSUMED INPUT ADDRESS IS A PSEUDO-ADDRESS. ADDR NUMBER ALTERED: HYPHEN INSERTED INPUT ADDRESS IS A PSEUDO-ADDRESS. ADDR NUMBER ALTERED: HYPHEN DELETED
	A	1A,BL	LOT HAS MORE ITEMS THAN LISTED
	B	1A	LOT HAS MORE ITEMS THAN LISTED.ADDR NUMBER ALTERED: RANGE ASSUMED LOT HAS MORE ITEMS THAN LISTED.ADDR NUMBER ALTERED: HYPHEN INSERTED LOT HAS MORE ITEMS THAN LISTED.ADDR NUMBER ALTERED: HYPHEN DELETED

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
01 (cont)	C	1,1A,1E	IN MARBLE HILL - LEGAL BORO IS MANHATTAN IN MARBLE HILL - LEGAL BORO IS MANHATTAN. ADDR NUMBER ALTERED: RANGE ASSUMED IN MARBLE HILL - LEGAL BORO IS MANHATTAN. ADDR NUMBER ALTERED: HYPHEN INSERTED IN MARBLE HILL - LEGAL BORO IS MANHATTAN. ADDR NUMBER ALTERED: HYPHEN DELETED
			ON RIKERS ISL - LEGAL BORO IS THE BRONX ON RIKERS ISL - LEGAL BORO IS THE BRONX. ADDR NUMBER ALTERED: RANGE ASSUMED ON RIKERS ISL - LEGAL BORO IS THE BRONX. ADDR NUMBER ALTERED: HYPHEN INSERTED ON RIKERS ISL - LEGAL BORO IS THE BRONX. ADDR NUMBER ALTERED: HYPHEN DELETED
	D	1*,2,3*	PARTIAL STREET NAME USED TO MEET SNL REQUIREMENT
	E	1,1E	OUTPUT ADDRESS RANGE IS SPLIT BY SCHOOL DISTRICT BOUNDARY
		1E	OUTPUT ADDRESS RANGE IS SPLIT BY ELECTION DISTRICT BOUNDARY
			OUTPUT ADDRESS RANGE IS SPLIT BY SCHOOL & ELECTION DISTRICT BOUNDARIES
	F	BN	THIS BIN IS TEMPORARY AND WILL BE REPLACED IN THE FUTURE
	G	1,1A,1E	ADDR NUMBER ALTERED: RANGE ASSUMED. NOTE: INCONSISTENT ODD/EVEN ADDR RANGE
	H	2,3S	THESE STREETS INTERSECT ONCE-COMPASS DIRECTION IGNORED
	I	1,1A,1E	INPUT IS NON-ADDRESSABLE PLACE NAME - ADDRESS NUMBER IGNORED

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES
(As of Geosupport Software Version 10.1)

GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
01 (cont)	J [not imple-mented	1,1A,1E, 2,3*	<Full street name including EAST or WEST as first word> ASSUMED [An input Bronx or Manhattan street name is missing EAST or WEST as its first word, and the intended full street name is unambiguous]
		2,3*	<Full street name> AND <other full street name> ASSUMED [Two input Bronx or Manhattan street names are missing EAST or WEST as their first words, and the intended names are unambiguous]
		3*	<Full street name>, <second full street name> AND <third full street name> ASSUMED [Three input Bronx or Manhattan street names are missing EAST or WEST as their first words, and the intended names are unambiguous]
	K	1,1A,1E	EMBEDDED BLANK IN ADDRESS NUMBER HAS BEEN REPLACED WITH A HYPHEN
	L or R	3,3C	<LEFT or RIGHT> SIDE OF SEGMENT IS IN <BROOKLYN or QUEENS> or <LEFT or RIGHT> SIDE OF SEGMENT IS IN <NASSAU or WESTCHESTER> - NO INFO RETURNED FOR THAT SIDE
	M	1,1A,1E	INPUT ADDRESS NUMBER IS ZERO
	N	1,1A,1E, 2,3*,D*	STREET NAME(S) AND STREET CODE(S) BOTH SPECIFIED AS INPUT - <CODE(S) or NAMES> IGNORED
	O	1,1A,1E	CAUTION: <BLOCK FACE or ADDR RANGE> CONTAINS OUT-OF-SEQUENCE AND/OR OPPOSITE PARITY ADDRESSES
	P	1,1E	IRREGULARLY CURVED STREET SEGMENT - SPATIAL COORDINATES RETURNED AS BLANKS

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
01 (cont)	Q	3	THESE STREETS INVOLVE A 'DOGLEG' - SHORTEST STRETCH PROVIDED
	S	1,1A,1E	<HNI or HNS> DISPLAY ADDRESS NUMBER BOTH SPECIFIED AS INPUT-<HNI or HNS> IGNORED
	T	2	NON-INTERSECTION NAME IGNORED
	U	3S	STRETCH HAS MORE ITEMS THAN LISTED
	V	1,1E	<Normalized input address number> <Norm'd input street name> IS ON <LEFT or RIGHT> SIDE OF <True street name> [This warning is issued for vanity addresses, NAPs other than complexes (for which an underlying address is not available), and certain alternative addresses known as type 'B' addresses.] or <Address number> <Street name> IS THE UNDERLYING ADDRESS OF <Normalized input NAP> [This warning is issued for NAPs other than complexes, for which an underlying address is available.]
[GRCs greater than 01 are rejects or errors]			
02		2	THESE STREETS INTERSECT TWICE-COMPASS DIRECTION REQUIRED
03	3 thru 9	2	THESE STREETS INTERSECT MORE THAN TWICE-CANNOT BE PROCESSED [Reason Code value indicates number of times the streets intersect. The value '9' signifies 9 or more.]
04		1A,BL	1A/BL VERSION SWITCH INVALID - MUST BE S. ONLY STANDARD IS SUPPORTED

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
05		3S	FOR FUNCTION 3S, ONLY FIRST BOROUGH CODE IS PERMITTED
07		1,1A,1E	FOR A NAME OF A COMPLEX, 5-DIGIT STREET CODE INPUT IS NOT PERMITTED
08		All but B*	INVALID STREET NAME NORMALIZATION FORMAT FLAG - MUST BE BLANK, C OR S
09		3C	<Compass direction> SIDE OF STREET SEGMENT IS NOT IN <borough name>
10		All but B*	INVALID SNL VALUE - MUST BE BETWEEN 4 AND 32 INCLUSIVE
11	0	1*,2,3*	<Street name> NOT RECOGNIZED. THERE ARE NO SIMILAR NAMES *As of Version 10.0 this message is used for batch in addition to CICS.*
12		2	INTERSECTION NAME NOT FOUND
13	1	1,1A,1E	ADDRESS NBR <value> CONTAINS AN INVALID CHARACTER <character> IN POSITION <position number>
	2	1,1A,1E	ADDRESS NBR <value> HAS MORE THAN 3 DIGITS AFTER DASH
	3	1,1A,1E	ADDRESS NBR <value> HAS TOO MANY DASHES
	4	1,1A,1E	ADDRESS NBR <value> HAS NO DIGITS AFTER THE DASH
	6	1,1A,1E	ADDRESS NBR <value> HAS TOO MANY DIGITS (MORE THAN 5)
	7	1,1A,1E	ADDRESS NBR <value> IS NOT COMPLETE AS ENTERED
	8	1,1A,1E	ADDRESS NBR <value> - PORTION AFTER HYPHEN EXCEEDS ALLOWABLE MAXIMUM

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES
(As of Geosupport Software Version 10.1)

GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
13 (cont	9	1,1A,1E, D*	ADDRESS NBR <hse nr value> INVALID INTERNAL FORMAT
	A	1,1A,1E, D*	ADDRESS NBR <value> HAS AN UNKNOWN OR INVALID SUFFIX/ENDING
	B	1,1A,1E	INPUT CONTAINS NO ADDRESS NUMBER
	C	1,1A,1E	ADDRESS NBR <value> HAS AN EMBEDDED BLANK
	D	1,1A,1E	ADDRESS NBR HAS INVALID FORMAT FOR EDGEWATER PARK
	E	1,1A,1E	THIS STREET HAS HYPHENATED ADDRESS NBRS ONLY. TRY <address nbr with hyphen inserted to left of penultimate digit> OR <address nbr with hyphen inserted to left of plusquepenultimate digit>
	F	1,1A,1E	THIS STREET HAS UNHYPHENATED ADDRESS NBRS ONLY. TRY <digits of address number to left of dash only> OR <digits to left and right of dash concatenated without the dash>
	G	1,1A,1E	ADDRESS NUMBER HAS INVALID HYPHENATION FOR THIS STREET [Input address number is an unhyphenated 2-digit number, but the input street has hyphenated address numbers only.]
14		3S	INPUT DOES NOT DEFINE A STREET STRETCH, SINCE INPUT INTERSECTIONS ARE IDENTICAL
15		All but B*	STREET NAME CANNOT BE NORMALIZED
16		1*	STREET NAME IS MISSING

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
17		All	BOROUGH CODE IS MISSING
18		BL	TAX BLOCK NOT NUMERIC
19		BL	TAX LOT NOT NUMERIC
20		BN	BUILDING IDENTIFICATION NUMBER (BIN) IS MISSING
21		BN	BUILDING IDENTIFICATION NUMBER (BIN) NOT FOUND
22		BN	INVALID BIN FORMAT: NON-NUMERIC, FIRST DIGIT NOT 1-5 OR REST OF DIGITS ALL ZERO
23		BN	TEMPORARY DEPARTMENT OF BUILDINGS BIN: EXISTS ONLY IN D.O.B FILES
24		3*	ON STREET IS MISSING
25		2,3*	CROSS STREET 1 IS MISSING
26		2,3*	CROSS STREET 2 IS MISSING
27		All	INVALID WORK AREA FORMAT INDICATOR - MUST BE C OR BLANK
28		1,1A,1E	A PARTIAL STREET NAME MAY NOT BE USED IN A FREE-FORM ADDRESS
29		1,1A,1E, 3*	INTERSECTION <INTERSECTION NAME> MAY NOT SERVE AS ON-STREET
30		2	<STREET NAME> IS NOT PART OF <INTERSECTION NAME>
31-37			*As of Version 10.0 GRC 31 through GRC 37 are replaced by GRC 11 and GRC EE.* *See descriptions of GRC 11 and GRC EE.*

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
38		3S	<Compass direction value> IS AN INVALID COMPASS DIRECTION VALUE FOR <FIRST or SECOND> INPUT INTERSECTION
39		2,3C	INVALID COMPASS DIRECTION VALUE - MUST BE N, S, E OR W
40		2,3C	COMPASS DIRECTION VALUE IS INVALID FOR THIS INPUT LOCATION
41		1,1A,1E	THIS STREET HAS NO ADDRESSES
42	blank	1,1A,1E	ADDRESS NUMBER OUT OF RANGE
	1	1,1A,1E	ADDRESS NUMBER OUT OF RANGE. CORRECT DIGITS OR INSERT HYPHEN AS <AB-CD> OR <A-BCD> [where input was of the form ABCD]
	2	1,1A,1E	ADDRESS NUMBER OUT OF RANGE. CORRECT DIGITS OR TRY <AB> OR <ABCD> [where input was of the form AB-CD]
44		3C	INPUT DOES NOT DEFINE A BLOCK FACE
45		3	INPUT DOES NOT DEFINE A STREET SEGMENT
46		3,3C	STREET COMBINATION NOT UNIQUE [The input is ambiguous, i.e., it describes more than one valid street segment.]
47		1,1A,1E COW Only	INVALID HNL VALUE - MUST BE BETWEEN 12 AND 16 INCLUSIVE
48		1,1A,1E COW Only	INVALID HOUSE NUMBER JUSTIFICATION VALUE - MUST BE L, R OR BLANK

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
49		1,1A,1E COW Only	ADDRESS NUMBER CANNOT BE NORMALIZED WITHIN REQUESTED HNL
50	1 thru 4	1,1A,1E, 2,3*	<Input street name> IS AN INVALID STREET NAME FOR THIS LOCATION [The Reason Code indicates the number of valid street names returned in the Similar Names list.]
51		1,1E,2,3, 3C	CROSS STREET NAMES FLAG MUST BE E OR BLANK
52		All but 1,1E,2,3, 3C	CROSS STREET NAMES OPTION IS INVALID FOR THIS FUNCTION
55		2,3*	NON-ADDRESSABLE PLACE NAME PROCESSING IS NOT AVAILABLE FOR THIS FUNCTION
56		1E	ADDRESS IS SPLIT AMONG MULTIPLE ELECTION DISTRICTS. ADDRESS NBR SUFFIX REQUIRED [The input address is associated with more than one Election District (ED). Function 1E requires an address number suffix to be included with this address to identify a portion of the building specific to one ED.]
57			*As of Version 10.0 GRC 57 is replaced by GRC 67. See description of GRC 67.*
58		1,1A,1E	NON-ADDRESSABLE PLACE NAME NOT FOUND
59		1*,2,3*	STREET NAME CANNOT BE NORMALIZED WITHIN REQUESTED SNL
61		3S	STREET STRETCH NOT FOUND

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES
(As of Geosupport Software Version 10.1)

GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
62		2,3S	<Street name> & <other street name> DO NOT INTERSECT
64		1,1A,1E, 2,3*,D*	STREET CODE NOT FOUND
65		1,1E	INVALID ROADBED REQUEST SWITCH. MUST BE R OR BLANK
66		3S	<Street name> & <other street name> INTERSECT MORE THAN TWICE-CANNOT BE PROCESSED
67	E,G, P,R, S,T	All batch only	ERROR ACCESSING GEOSUPPORT FILE: <file name> NOTIFY SYSTEM SUPPORT [This can be an installation error or a system error Notify System Support.] *As of Version 10.0 this message is used for CICS in addition to batch.*
68		3S	<Street name> & <other street name> INTERSECT TWICE-COMPASS DIRECTION REQ'D
73		1A,BL	LEGACY VERSION OF FUNCTIONS 1A AND BL IS DISCONTINUED. SEE TECH BULLETIN 05-1
75		1,1A,1E	DUPLICATE ADDRESS-USE <pseudo-streetname1> OR <pseudo-streetname2>
76		All but 1,1E	ROADBED REQUEST SWITCH NOT IMPLEMENTED FOR THIS FUNCTION
77		BL	TAX LOT NOT FOUND
88	blank	All	GEOSUPPORT SYSTEM ERROR. NOTIFY DCP/GSS USER SUPPORT [An internal Geosupport problem, not a user error.]

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES (As of Geosupport Software Version 10.1)			
GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
88 (cont)	1-8, C-H	All	GEOSUPPORT SYSTEM ERROR. NOTIFY DCP/GSS USER SUPPORT & REPORT REASON CODE = <value> [An internal Geosupport problem, not a user error.]
	9	All,CICS only	CICS ERROR. NOTIFY DATA CENTER TECHNICAL SUPPORT [A system error, not a user error.]
	A	All	MODULE HAS NOT LOADED. NOTIFY TECHNICAL SUPPORT [A system error, not a user error.]
	B	All	SYSTEM ERROR. NOTIFY TECHNICAL SUPPORT & REPORT REASON CODE = B [An internal Geosupport problem or a system error, not user error.]
89		2,3C,3S, BN	LONG WORK-AREA-2 OPTION IS INVALID FOR THIS FUNCTION
		1,1E,3 COW Only	
90		1,1A,1E, 3,BL	LONG WORK-AREA-2 FLAG MUST BE L OR BLANK
96		All, CICS only	AN I/O ERROR HAS OCCURRED. TRY AGAIN
97		BB,BF	INPUT IS BEYOND THE LAST STREET NAME IN THE BOROUGH OF <borough name>
98		All	NO INPUT DATA RECEIVED
99		All	INVALID BOROUGH CODE. MUST BE 1, 2, 3, 4 OR 5

GEOSUPPORT SYSTEM RETURN CODES, REASON CODES AND MESSAGES
(As of Geosupport Software Version 10.1)

GRC	REASON CODE	FUNCTIONS (* = wildcard)	MESSAGE (LITERAL TEXT IN UPPERCASE, <Variable values in angled brackets>, [Comments in Square Brackets & Mixed Case])
EE	1	1*,2,3*	<Street name> NOT RECOGNIZED. IS IT <similar street name>? [Issued when there is precisely one similar name.] *As of Version 10.0 this message is used for batch in addition to CICS.*
	2 thru 9, A	1*,2,3*	<Street name> NOT RECOGNIZED. THERE ARE <number> SIMILAR NAMES [Issued when there is more than one similar name. Reason Code indicates number of similar names. Reason Code 'A' signifies 10 similar names. The similar names are returned in WA1.] *As of Version 10.0 this message is used for batch in addition to CICS.*
??		N/A	INVALID FUNCTION CODE

APPENDIX 5: GEOSUPPORT COPY FILES (MSW)

This appendix contains printouts of the Geosupport MSW COPY files for COBOL, Assembler, PL/1, C and NATURAL. (For C, COPY files take the form of header files. For NATURAL, COPY files take the form of Local Data Areas.)

The Geosupport COPY files contain source code layouts of the Geosupport work areas. These files are stored in a COPY library that can be accessed by user application programs at compile time. Each supported programming language has an appropriate declarative statement for referencing COPY files at compile time. The Geosupport COPY files are listed below.

GEOSUPPORT SYSTEM COPY FILES (MSW)

MSW WORK AREA	FUNCTION(S)	LENGTH (bytes)	- - - - - - - - - - - - - COPY File Name - - - - - - - - - - - - - -				
			COBOL	ASSEMBLER	PL/1	C	NATURAL
WA1	All	884	W1COB	W1BAL	W1PL1	WAC	GEOLW1
WA2	1 (regular WA2), 1E (regular WA2), 2, 3 (regular WA2), 3C	200	W2COB	W2BAL	W2PL1	WAC	GEOLW2
WA2	1 (long WA2), 1E (long WA2), 3 (long WA2)	300	W2COBL	W2BALL	W2PL1L	WAC	GEOLW2L
WA2	1A&BL (regular WA2),BN (*)	939	W2COB1A	W2BAL1A	W2PL11A	WAC	GEOLW21A
WA2	1A&BL (long WA2) (**)	17,683	W2COB1AL	W2BAL1AL	W2PL11AL	WAC	GEOLW2AL
WA2	3S	4,224	W2COB3S	W2BAL3S	W2PL13S	WAC	GEOLW23S

(*) Functions 1A, BL and BN share a single regular WA2 layout.

(**) Functions 1A and BL share a single long WA2 layout. (Function BN does not have the long WA2 option.)

See Section VIII.4 for a detailed discussion of the Geosupport COPY feature.

COBOL COPY Files (MSW)

W1COB COPY File

```
********************************************************************** 00000100
*****  THIS IS GEOSUPPORT SYSTEM COPY FILE W1COB, CONTAINING   ***** 00000200
*****  LAYOUT OF WORK AREA 1.       COPYLIB2       04/07/98   ***** 00000300
********************************************************************** 00000400
*****               INPUT FIELDS                              ***** 00000500
********************************************************************** 00000600
         05  GEO-WA1-IN-FUNCTION-CODE.                                 00000700
            10  GEO-WA1-IN-FUNCTION-1           PIC X.                 00000800
            10  GEO-WA1-IN-FUNCTION-2           PIC X.                 00000900
         05  GEO-WA1-IN-BORO                    PIC X.                 00001000
   ** NOTE  GEO-WA1-IN-HOUSENUM - HIGH HSE# INPUT IF FUNC 5            00001100
         05  GEO-WA1-IN-HOUSENUM                PIC X(12).             00001200
   ** NOTE  GEO-WA1-IN-HOUSENUM-INTERNAL - HIGH HSE# INPUT IF FUN5     00001300
         05  GEO-WA1-IN-HOUSENUM-INTERNAL       PIC X(6).              00001400
         05  GEO-WA1-IN-STREET-1                PIC X(32).             00001500
         05  GEO-WA1-IN-STREET-2                PIC X(32).             00001600
         05  GEO-WA1-IN-STREET-3                PIC X(32).             00001700
         05  GEO-WA1-IN-COMPASS                 PIC X.                 00001802
         05  GEO-WA1-IN-COMPASS2                PIC X.                 00001904
         05  GEO-WA1-IN-STREETCODE-1            PIC S9(6) COMP-3.      00002104
         05  GEO-WA1-IN-STREETCODE-2            PIC S9(6) COMP-3.      00002204
         05  GEO-WA1-IN-STREETCODE-3            PIC S9(6) COMP-3.      00002304
         05  GEO-WA1-IN-ROADBED-REQ-SWITCH      PIC X.                 00002508
         05  GEO-WA1-IN-BORO-2                  PIC X.                 00002604
         05  GEO-WA1-IN-BORO-3                  PIC X.                 00002704
         05  GEO-WA1-IN-SNL                     PIC X(2).              00002804
         05  GEO-WA1-IN-10SC-1                  PIC X(11).             00002904
         05  GEO-WA1-IN-10SC-2                  PIC X(11).             00003004
         05  GEO-WA1-IN-10SC-3                  PIC X(11).             00003104
   ** NOTE: GEO-WA1-IN-CUI       NOT IMPLEMENTED   **                  00003204
         05  GEO-WA1-IN-CUI                     PIC X(5).              00003304
         05  GEO-WA1-IN-BBL.                                           00003404
            10  GEO-WA1-IN-BL-BORO              PIC X.                 00003504
            10  GEO-WA1-IN-BLOCKNUM             PIC X(5).              00003604
            10  GEO-WA1-IN-LOTNUM               PIC X(4).              00003704
         05  FILLER                             PIC X.                 00003804
         05  GEO-WA1-IN-BIN                     PIC X(7).              00004004
   *****************************************************************  00004104
   ** NOTE: TO REQUEST COMPACT NAMES OPTION, SET                 **  00004204
   **       GEO-WA1-IN-COMPACT-NAME-FLAG TO "C".                 **  00004304
   ** NOTE: TO REQUEST THE LONG WORKAREA 2, SET                  **  00004404
   **       GEO-WA1-IN-LONG-WORKAREA2-FLAG TO "L".  AT PRESENT,  **  00004504
   **       ONLY FUNCTION 3 HAS THE LONG WA2 OPTION.             **  00004604
   ** NOTE: IF APPLICATION IS RUNNING ON A NON-IBM MAIN FRAME,   **  00004704
   **       SET GEO-WA1-IN-NON-IBM-MAIN-FRAME EQUAL TO "X"       **  00004804
   ** NOTE: FOR FUNCTIONS 1A AND BL, TO REQUEST THE STANDARD     **  00004904
   **       WORKAREA2 FORMAT, SET GEO-WA1-IN-1ABL-VERSION TO "S" **  00005004
   ** NOTE: TO REQUEST THE LEGACY WORKAREA2 FORMAT,              **  00005104
   **       SET GEO-WA1-IN-1ABL-VERSION TO " " OR "L".           **  00005204
   *****************************************************************  00005704
         05  GEO-WA1-IN-COMPACT-NAME-FLAG       PIC X.                 00005804
         05  GEO-WA1-IN-LONG-WORKAREA2-FLAG     PIC X.                 00005904
         05  GEO-WA1-IN-LOW-HOUSENUM            PIC X(12).             00006004
         05  GEO-WA1-IN-LOW-HSENUM-INTERNAL     PIC X(6).              00006104
         05  GEO-WA1-IN-NON-IBM-MAIN-FRAME      PIC X(1).              00006204
         05  GEO-WA1-IN-1ABL-VERSION            PIC X(1).              00006304
         05  GEO-WA1-IN-XSTREET-FLAG            PIC X(1).              00006404
         05  GEO-WA1-IN-NONEED-STNAME-FLAG      PIC X(1).              00006504
         05  FILLER                             PIC X(3).              00006604
********************************************************************** 00006704
*****               OUTPUT FIELDS                             ***** 00006804
********************************************************************** 00006904
         05  GEO-WA1-OUT-LOW-HOUSENUM           PIC X(12).             00007004
         05  GEO-WA1-OUT-BORONAME               PIC X(9).              00007104
         05  GEO-WA1-OUT-STREET-1               PIC X(32).             00007204
```

W1COB COPY File (continued)

```
           05  GEO-WA1-OUT-STREET-2                PIC X(32).          00007304
           05  GEO-WA1-OUT-STREET-3                PIC X(32).          00007404
           05  GEO-WA1-OUT-HOUSENUM                PIC X(12).          00007504
           05  GEO-WA1-OUT-HOUSENUM-INTERNAL       PIC X(6).           00007604
           05  GEO-WA1-OUT-HOUSE-INT-FORMAT REDEFINES                  00007704
               GEO-WA1-OUT-HOUSENUM-INTERNAL.                          00007804
               10  GEO-WA1-OUT-HOUSE-INT-PACKED    PIC X(5).           00007904
               10  GEO-WA1-OUT-HOUSE-INT-SUFFIX    PIC X.              00008004
           05  FILLER                              PIC X(7).           00008104
           05  GEO-WA1-OUT-PB5SC-1                 PIC S9(6) COMP-3.   00008204
           05  GEO-WA1-OUT-PB-5SC-1 REDEFINES GEO-WA1-OUT-PB5SC-1.     00008304
               10  FILLER                          PIC X(1).           00008404
               10  GEO-WA1-OUT-STREETCODE-1-KEY    PIC S9(5) COMP-3.   00008504
           05  FILLER                              PIC X(2).           00008604
           05  GEO-WA1-OUT-PB5SC-2                 PIC S9(6) COMP-3.   00008704
           05  GEO-WA1-OUT-PB-5SC-2 REDEFINES GEO-WA1-OUT-PB5SC-2.     00008804
               10  FILLER                          PIC X(1).           00008904
               10  GEO-WA1-OUT-STREETCODE-2-KEY    PIC S9(5) COMP-3.   00009004
           05  FILLER                              PIC X(2).           00009104
           05  GEO-WA1-OUT-PB5SC-3                 PIC S9(6) COMP-3.   00009204
           05  GEO-WA1-OUT-PB-5SC-3 REDEFINES GEO-WA1-OUT-PB5SC-3.     00009304
               10  FILLER                          PIC X(1).           00009404
               10  GEO-WA1-OUT-STREETCODE-3-KEY    PIC S9(5) COMP-3.   00009504
           05  FILLER                              PIC X(3).           00009604
           05  GEO-WA1-BROWSE                      PIC X(40).          00009704
           05  GEO-WA1-OUT-10SC-1                  PIC X(11).          00009804
           05  GEO-WA1-OUT-10SC-2                  PIC X(11).          00009904
           05  GEO-WA1-OUT-10SC-3                  PIC X(11).          00010004
           05  GEO-WA1-OUT-CUI                     PIC X(5).           00010104
      ** NOTE: GEO-WA1-OUT-CUI       NOT IMPLEMENTED   **              00010204
           05  GEO-WA1-OUT-BBL.                                        00010304
              10  GEO-WA1-OUT-BL-BORO              PIC X.              00010404
              10  GEO-WA1-OUT-BLOCKNUM             PIC X(5).           00010504
              10  GEO-WA1-OUT-LOTNUM               PIC X(4).           00010604
           05  FILLER                              PIC X.              00010704
           05  GEO-WA1-OUT-BIN                     PIC X(7).           00010806
           05  GEO-WA1-OUT-SND-ATTR                PIC X.              00011304
           05  GEO-WA1-OUT-REASON-CODE             PIC X.              00011404
           05  FILLER                              PIC X(2).           00011504
           05  GEO-WA1-OUT-RETURN-CODE.                                00011604
              10  GEO-WA1-OUT-RC-1                 PIC X.              00011704
              10  GEO-WA1-OUT-RC-2                 PIC X.              00011804
           05  GEO-WA1-OUT-ERROR-MESSAGE           PIC X(80).          00011904
           05  GEO-WA1-OUT-NUM-SIMILAR-NAMES       PIC S999 COMP-3.    00012004
           05  GEO-WA1-OUT-SIMILAR-NAMES           PIC X(32)           00012104
                                                   OCCURS 10 TIMES.    00013002
```

W2COB COPY File

```
*****************************************************************  00000100
***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2COB, CONTAINING THE ***  00000200
***  LAYOUT OF WORK AREA 2 FOR FUNCTIONS: 1, 1E, 2, 2C, 3, 3C, ***  00000300
***  5. PLEASE NOTE THAT FUNCTIONS 2 AND 2C SHARE A SINGLE WORK***  00000400
***  AREA 2 LAYOUT.                                   04/03/01 ***  00000513
*****************************************************************  00000600
***  LAST MODIFIED  05/19/06                                   ***  00000731
*****************************************************************  00000923
       05  GEO-WA2-FUNCTION1.                                       00001023
          10  FILLER                          PIC X(21).            00001123
          10  GEO-WA2-FN1-CONT-PARITY         PIC X.                00001223
          10  GEO-WA2-FN1-LOW-HOUSENUM-INT.                         00001323
             15  GEO-WA2-FN1-LOW-HOUSENUM     PIC X(5).             00001423
             15  GEO-WA2-FN1-LOW-HOUSENUMSFX  PIC X.                00001523
                                                                    00001626
          10  GEO-WA2-FN1-HI-HOUSENUM-INT.                          00001723
             15  GEO-WA2-FN1-HI-HOUSENUM      PIC X(5).             00001823
             15  GEO-WA2-FN1-HI-HOUSENUMSFX   PIC X.                00001923
                                                                    00002026
          10  GEO-WA2-FN1-ALX                 PIC X.                00002131
          10  GEO-WA2-FN1-NUM-X-ST-LOW-END    PIC X.                00002223
          10  GEO-WA2-FN1-LOW-PBSC            PIC S9(7) COMP-3      00002323
                                              OCCURS 5 TIMES.       00002423
          10  GEO-WA2-FN1-NUM-X-ST-HI-END     PIC X.                00002523
          10  GEO-WA2-FN1-HI-PBSC             PIC S9(7) COMP-3      00002623
                                              OCCURS 5 TIMES.       00002723
          10  GEO-WA2-FN1-COMDIST.                                  00002823
              15  GEO-WA2-FN1-COMDIST-BORO    PIC X.                00002928
              15  GEO-WA2-FN1-COMDIST-NUMBER  PIC X(2).             00003023
          10  GEO-WA2-FN1-ZIP                 PIC X(5).             00003123
          10  GEO-WA2-FN1-SLA                 PIC X.                00003223
          10  GEO-WA2-FN1-HCD                 PIC X(2).             00003323
          10  GEO-WA2-FN1-SOS                 PIC X.                00003423
          10  GEO-WA2-FN1-CONT-PARITY-IND     PIC X.                00003523
          10  GEO-WA2-FN1-2000-CENS-TRCT      PIC X(6).             00003623
          10  GEO-WA2-FN1-2000-CENS-BLK       PIC X(4).             00003723
          10  GEO-WA2-FN1-INSTRUC-DIV         PIC X(2).             00003823
          10  FILLER                          PIC X(2).             00003923
          10  GEO-WA2-FN1-HEALTHAREA          PIC X(4).             00004023
          10  GEO-WA2-FN1-SANI-REC            PIC X(3).             00004123
          10  GEO-WA2-FN1-FEATURE-TYPE        PIC X.                00004223
          10  GEO-WA2-FN1-RESDCP              PIC X.                00004328
          10  GEO-WA2-FN1-CURVE-FLAG          PIC X.                00004428
          10  GEO-WA2-FN1-POLICEDIST.                               00004523
             15  GEO-WA2-FN1-POL-PATR-BORO-CMD  PIC X.              00004628
             15  GEO-WA2-FN1-POL-PRECINCT     PIC X(3).             00004723
          10  GEO-WA2-FN1-SCHOOLDIST          PIC X(2).             00004823
          10  FILLER                          PIC X(15).            00004928
          10  GEO-WA2-FN1-SEGMENT-TYPE        PIC X.                00005028
          10  GEO-WA2-FN1-SANIDIST.                                 00005123
              15  GEO-WA2-FN1-SANIDIST-BORO   PIC X.                00005228
              15  GEO-WA2-FN1-SANIDIST-NUMBER PIC X(2).             00005323
          10  GEO-WA2-FN1-SANITATION-SUBSEC   PIC X(2).             00005423
  ** NOTE:10  GEO-WA2-FN1-FIRESEC ==> FIRE DIVISION **              00005523
          10  GEO-WA2-FN1-FIRESEC             PIC X(2).             00005623
          10  GEO-WA2-FN1-FIREBAT             PIC X(2).             00005723
          10  GEO-WA2-FN1-FIRECO.                                   00005823
              15  GEO-WA2-FN1-FIRECO-TYPE     PIC X.                00005928
              15  GEO-WA2-FN1-FIRECO-NUM      PIC X(3).             00006023
          10  GEO-WA2-FN1-SPECIAL-ADDR-FLAG   PIC X.                00006128
          10  GEO-WA2-FN1-MARBLE-RIKER-FLAG   PIC X.                00006228
          10  GEO-WA2-FN1-SPLIT-SCHOOL-FLAG   PIC X.                00006323
          10  GEO-WA2-FN1-PREFERRED-LGC       PIC X(2).             00006423
          10  GEO-WA2-FN1-LIONFACECODE        PIC X(4).             00006523
          10  GEO-WA2-FN1-LIONSEQ             PIC X(5).             00006623
```

W2COB COPY File (continued)

```
             10  GEO-WA2-FN1-1990-CENSUSTRACT      PIC X(6).            00006723
             10  FILLER                            PIC X(4).            00006823
             10  GEO-WA2-FN1-DYN-BLOCK             PIC X(3).            00006911
             10  GEO-WA2-FN1-XCOORD                PIC X(7).            00007011
             10  GEO-WA2-FN1-YCOORD                PIC X(7).            00007111
             10  GEO-WA2-FN1-SEGMENTLENGTH         PIC X(5).            00007211
             10  GEO-WA2-FN1-SANI-REG              PIC X(5).            00007306
          05  GEO-WA2-FUNCTION2   REDEFINES GEO-WA2-FUNCTION1.          00007400
             10  FILLER                            PIC X(21).           00007500
             10  GEO-WA2-FN2-DUPINTERFLAG          PIC X.               00007600
             10  FILLER                            PIC X(9).            00007700
             10  GEO-WA2-FN2-PREFERRED-LGC1        PIC X(2).            00007800
             10  GEO-WA2-FN2-PREFERRED-LGC2        PIC X(2).            00007900
             10  GEO-WA2-FN2-NUM-OF-INTERSECTS     PIC X.               00008000
             10  GEO-WA2-FN2-INTERSECT-PBSC        PIC 9(7) COMP-3      00008100
                                                   OCCURS 5 TIMES.      00008200
             10  GEO-WA2-FN2-COMPDIR               PIC X.               00008305
             10  GEO-WA2-FN2-LEVEL-LIST OCCURS 5 TIMES.                 00008519
                 15 GEO-WA2-FN2-LEVEL-CODES                             00008619
                    OCCURS 2 TIMES                 PIC X.               00008719
             10  GEO-WA2-FN2-INSTRUC-DIV           PIC X(2).            00008818
     ** NOTE:10  GEO-WA2-FN2-FIRESEC ==> FIRE DIVISION **               00008918
             10  GEO-WA2-FN2-FIRESEC               PIC X(2).            00009018
             10  GEO-WA2-FN2-FIREBAT               PIC X(2).            00009118
             10  GEO-WA2-FN2-FIRECO.                                    00009218
                 15  GEO-WA2-FN2-FIRECO-TYPE       PIC X.               00009328
                 15  GEO-WA2-FN2-FIRECO-NUM        PIC X(3).            00009418
             10  GEO-WA2-FN2-COMDIST.                                   00009518
                 15  GEO-WA2-FN2-COMDIST-BORO      PIC X.               00009628
                 15  GEO-WA2-FN2-COMDIST-NUMBER    PIC X(2).            00009718
             10  GEO-WA2-FN2-ZIP                   PIC X(5).            00009818
             10  GEO-WA2-FN2-SLA                   PIC X.               00009918
             10  GEO-WA2-FN2-2000-CENS-TRCT        PIC X(6).            00010018
             10  FILLER                            PIC X(3).            00010101
             10  GEO-WA2-FN2-HEALTHAREA            PIC X(4).            00010201
             10  FILLER                            PIC X(9).            00010303
             10  GEO-WA2-FN2-NODE-NUM              PIC X(7).            00010403
             10  GEO-WA2-FN2-XCOORD                PIC X(7).            00010501
             10  GEO-WA2-FN2-YCOORD                PIC X(7).            00010601
             10  FILLER                            PIC X(4).            00010801
             10  GEO-WA2-FN2-POLICEDIST.                                00010901
                15  GEO-WA2-FN2-POL-PATR-BORO-CMD  PIC X.               00011028
                15  GEO-WA2-FN2-POL-PRECINCT       PIC X(3).            00011101
             10  GEO-WA2-FN2-SCHOOLDIST            PIC X(2).            00011201
             10  GEO-WA2-FN2-MARBLE-RIKER-FLAG     PIC X.               00011328
             10  GEO-WA2-FN2-1990-CENSUSTRACT      PIC X(6).            00011401
             10  GEO-WA2-FN2-SANBORN1-BVOLPAGE.                         00011501
                15  GEO-WA2-FN2-SANBORN1-BORO      PIC X.               00011628
                15  GEO-WA2-FN2-SANBORN1-VOL-NUM   PIC X(3).            00011701
                15  GEO-WA2-FN2-SANBORN1-PAGE-NUM  PIC X(4).            00011801
             10  GEO-WA2-FN2-SANBORN2-BVOLPAGE.                         00011901
                15  GEO-WA2-FN2-SANBORN2-BORO      PIC X.               00012028
                15  GEO-WA2-FN2-SANBORN2-VOL-NUM   PIC X(3).            00012101
                15  GEO-WA2-FN2-SANBORN2-PAGE-NUM  PIC X(4).            00012201
             10  FILLER                            PIC X(38).           00012301
          05  GEO-WA2-FUNCTION3   REDEFINES GEO-WA2-FUNCTION1.          00012401
             10  FILLER                            PIC X(21).           00012501
             10  GEO-WA2-FN3-DUP-KEY-FLAG          PIC X.               00012601
             10  GEO-WA2-FN3-CURVE-FLAG            PIC X.               00012811
             10  GEO-WA2-FN3-LOCATION-STATUS       PIC X.               00012911
             10  GEO-WA2-FN3-COUNTY-BOUNDARY       PIC X.               00013011
             10  FILLER                            PIC X(4).            00013111
             10  GEO-WA2-FN3-PREFERRED-LGC1        PIC X(2).            00013211
             10  GEO-WA2-FN3-PREFERRED-LGC2        PIC X(2).            00013311
             10  GEO-WA2-FN3-PREFERRED-LGC3        PIC X(2).            00013411
             10  GEO-WA2-FN3-NUM-X-ST-LOW-END      PIC X.               00013511
```

W2COB COPY File (continued)

```
           10  GEO-WA2-FN3-LOW-PBSC                  PIC S9(7) COMP-3    00013611
                                                     OCCURS 5 TIMES.     00013711
           10  GEO-WA2-FN3-NUM-X-ST-HI-END           PIC X.              00013811
           10  GEO-WA2-FN3-HI-PBSC                   PIC S9(7) COMP-3    00013911
                                                     OCCURS 5 TIMES.     00014011
           10  GEO-WA2-FN3-SLA                       PIC X.              00014111
           10  GEO-WA2-FN3-REVERSALFLAG              PIC X.              00014211
           10  GEO-WA2-FN3-LEFT-COMDIST.                                 00014311
               15  GEO-WA2-FN3-LEFT-COMDIST-BORO PIC X.                  00014428
               15  GEO-WA2-FN3-LEFT-COMDIST-NUM  PIC X(2).               00014511
           10  GEO-WA2-FN3-RIGHT-COMDIST.                                00014611
               15 GEO-WA2-FN3-RIGHT-COMDIST-BORO PIC X.                  00014728
               15  GEO-WA2-FN3-RIGHT-COMDIST-NUM PIC X(2).               00014811
           10  GEO-WA2-FN3-LEFT-ZIP                  PIC X(5).           00014911
           10  GEO-WA2-FN3-RIGHT-ZIP                 PIC X(5).           00015011
           10  FILLER                                PIC X(18).          00015112
           10  GEO-WA2-FN3-LEFT-HEALTHAREA           PIC X(4).           00015711
           10  GEO-WA2-FN3-RIGHT-HEALTHAREA          PIC X(4).           00015811
           10  GEO-WA2-FN3-LEFT-INSTRUC-DIV          PIC X(2).           00015920
           10  GEO-WA2-FN3-RIGHT-INSTRUC-DIV         PIC X(2).           00016020
           10  GEO-WA2-FN3-LEFT-LOW-HOUSENUM         PIC X(7).           00016120
           10  GEO-WA2-FN3-LEFT-HI-HOUSENUM          PIC X(7).           00016220
           10  GEO-WA2-FN3-RIGHT-LOW-HOUSENUM        PIC X(7).           00016320
           10  GEO-WA2-FN3-RIGHT-HI-HOUSENUM         PIC X(7).           00016420
           10  GEO-WA2-FN3-CONT-PARITY-IND           PIC X.              00016520
           10  GEO-WA2-FN3-LIONFACECODE              PIC X(4).           00016620
           10  GEO-WA2-FN3-LIONSEQ                   PIC X(5).           00016720
           10  GEO-WA2-FN3-GENRECFLAG                PIC X.              00016820
           10  GEO-WA2-FN3-SEGMENTLENGTH             PIC S9(5) COMP-3.   00016920
           10  GEO-WA2-FN3-SEGMENTSLOPE              PIC X(3).           00017020
           10  GEO-WA2-FN3-SEGMENTORIENT             PIC X.              00017120
           10  FILLER                                PIC X(4).           00017220
           10  GEO-WA2-FN3-RESDCP                    PIC X(2).           00017320
           10  GEO-WA2-FN3-DOG-LEG                   PIC X.              00017420
           10  GEO-WA2-FN3-FEATURE-TYPE              PIC X.              00017521
           10  GEO-WA2-FN3-LEFT-POLDIST.                                 00017720
              15  GEO-WA2-FN3-L-POL-PATR-BOR-CMD PIC X.                  00017828
              15  GEO-WA2-FN3-L-POL-PRECINCT     PIC X(3).               00017920
           10  GEO-WA2-FN3-RIGHT-POLDIST.                                00018020
              15  GEO-WA2-FN3-R-POL-PATR-BOR-CMD PIC X.                  00018128
              15  GEO-WA2-FN3-R-POL-PRECINCT     PIC X(3).               00018220
           10  GEO-WA2-FN3-LEFT-SCHLDIST             PIC X(2).           00018320
           10  GEO-WA2-FN3-RIGHT-SCHLDIST            PIC X(2).           00018420
           10  GEO-WA2-FN3-MARBLE-RIKER-FLAG         PIC X.              00018528
           10  GEO-WA2-FN3-SEG-ID                    PIC X(7).           00018620
           10  GEO-WA2-FN3-SEGMENT-TYPE              PIC X.              00018729
        05  GEO-WA2-FUNCTION3C  REDEFINES GEO-WA2-FUNCTION1.             00018920
           10  GEO-WA2-FN3C-ACCESS-KEY               PIC X(21).          00019020
           10  GEO-WA2-FN3C-CURVE-FLAG               PIC X.              00019120
           10  GEO-WA2-FN3C-SEGMENT-TYPE             PIC X.              00019230
           10  GEO-WA2-FN3C-LOCATION-STATUS          PIC X.              00019430
           10  GEO-WA2-FN3C-COUNTY-BOUNDARY          PIC X.              00019530
           10  FILLER                                PIC X(4).           00019630
           10  GEO-WA2-FN3C-PREFERRED-LGC1           PIC X(2).           00019730
           10  GEO-WA2-FN3C-PREFERRED-LGC2           PIC X(2).           00019830
           10  GEO-WA2-FN3C-PREFERRED-LGC3           PIC X(2).           00019930
           10  GEO-WA2-FN3C-NUM-X-ST-LOW-END         PIC X.              00020030
           10  GEO-WA2-FN3C-LOW-PBSC                 PIC S9(7) COMP-3    00020130
                                                     OCCURS 5 TIMES.     00020230
           10  GEO-WA2-FN3C-NUM-X-ST-HI-END          PIC X.              00020330
           10  GEO-WA2-FN3C-HI-PBSC                  PIC S9(7) COMP-3    00020430
                                                     OCCURS 5 TIMES.     00020530
           10  GEO-WA2-FN3C-COMDIST.                                     00020630
               15  GEO-WA2-FN3C-COMDIST-BORO         PIC X.              00020730
               15  GEO-WA2-FN3C-COMDIST-NUMBER       PIC X(2).           00020830
           10  GEO-WA2-FN3C-ZIP                      PIC X(5).           00020930
```

W2COB COPY File (continued)

```
           10  GEO-WA2-FN3C-SLA                  PIC X.
           10  FILLER                            PIC X(7).
           10  GEO-WA2-FN3C-2000-CENS-TRCT       PIC X(6).
           10  GEO-WA2-FN3C-2000-CENS-BLK        PIC X(4).
           10  FILLER                            PIC X.
           10  GEO-WA2-FN3C-HEALTHAREA           PIC X(4).
           10  GEO-WA2-FN3C-REVERSALFLAG         PIC X.
           10  GEO-WA2-FN3C-SOS                  PIC X.
   ** NOTE:10  GEO-WA2-FN3C-FIRESEC ==> FIRE DIVISION **
           10  GEO-WA2-FN3C-FIRESEC              PIC X(2).
           10  GEO-WA2-FN3C-FIREBAT              PIC X(2).
           10  GEO-WA2-FN3C-FIRECO.
               15  GEO-WA2-FN3C-FIRECO-TYPE      PIC X.
               15  GEO-WA2-FN3C-FIRECO-NUM       PIC X(3).
           10  GEO-WA2-FN3C-SEG-ID               PIC X(7).
           10  GEO-WA2-FN3C-LOW-HOUSENUM         PIC X(7).
           10  GEO-WA2-FN3C-HI-HOUSENUM          PIC X(7).
           10  GEO-WA2-FN3C-LOW-HOUSENUM2        PIC X(7).
           10  GEO-WA2-FN3C-HI-HOUSENUM2         PIC X(7).
   * HOUSENUM2 ONLY PRESENT IF ODD & EVEN RANGES ARE ON
   * SAME SIDE OF STREET.
           10  GEO-WA2-FN3C-CONT-PARITY-IND      PIC X.
           10  GEO-WA2-FN3C-LIONFACECODE         PIC X(4).
           10  GEO-WA2-FN3C-LIONSEQ              PIC X(5).
           10  GEO-WA2-FN3C-GENRECFLAG           PIC X.
           10  GEO-WA2-FN3C-SEGMENTLENGTH        PIC S9(5) COMP-3.
           10  GEO-WA2-FN3C-SEGMENTSLOPE         PIC X(3).
           10  GEO-WA2-FN3C-SEGMENTORIENT        PIC X.
           10  GEO-WA2-FN3C-INSTRUC-DIV          PIC XX.
           10  GEO-WA2-FN3C-RESDCP               PIC X.
           10  GEO-WA2-FN3C-FEATURE-TYPE         PIC X.
           10  GEO-WA2-FN3C-POLICEDIST.
              15  GEO-WA2-FN3C-POL-PATR-BORO-CMD PIC X.
              15  GEO-WA2-FN3C-POL-PRECINCT      PIC X(3).
           10  GEO-WA2-FN3C-SCHOOLDIST           PIC X(2).
           10  GEO-WA2-FN3C-MARBLE-RIKER-FLAG    PIC X.
           10  GEO-WA2-FN3C-1990-CENSUSTRACT     PIC X(6).
           10  FILLER                            PIC X(4).
           10  GEO-WA2-FN3C-DYN-BLOCK            PIC X(3).
           10  FILLER                            PIC X(5).
        05  GEO-WA2-FUNCTION1E  REDEFINES GEO-WA2-FUNCTION1.
           10  FILLER                            PIC X(21).
           10  GEO-WA2-FN1E-CONT-PARITY          PIC X.
           10  GEO-WA2-FN1E-LOW-HOUSENUM-INT.
               15  GEO-WA2-FN1E-LOW-HOUSENUM     PIC X(5).
               15  GEO-WA2-FN1E-LOW-HSENUMSFX    PIC X.
           10  GEO-WA2-FN1E-HI-HOUSENUM-INT.
               15  GEO-WA2-FN1E-HI-HOUSENUM      PIC X(5).
               15  GEO-WA2-FN1E-HI-HSENUMSFX     PIC X.
           10  FILLER                            PIC X.
           10  GEO-WA2-FN1E-NUM-X-ST-LOW-END     PIC X.
           10  GEO-WA2-FN1E-LOW-PBSC             PIC S9(7) COMP-3
                                                 OCCURS 5 TIMES.
           10  GEO-WA2-FN1E-NUM-X-ST-HI-END      PIC X.
           10  GEO-WA2-FN1E-HI-PBSC              PIC S9(7) COMP-3
                                                 OCCURS 5 TIMES.
           10  GEO-WA2-FN1E-COMDIST.
               15  GEO-WA2-FN1E-COMDIST-BORO     PIC X.
               15  GEO-WA2-FN1E-COMDIST-NUMBER   PIC X(2).
           10  GEO-WA2-FN1E-ZIP                  PIC X(5).
           10  GEO-WA2-FN1E-SLA                  PIC X.
           10  GEO-WA2-FN1E-HCD                  PIC X(2).
           10  GEO-WA2-FN1E-SOS                  PIC X.
           10  GEO-WA2-FN1E-CONT-PARITY-IND      PIC X.
           10  GEO-WA2-FN1E-2000-CENS-TRCT       PIC X(6).
           10  GEO-WA2-FN1E-2000-CENS-BLK        PIC X(4).
```

W2COB COPY File (continued)

```
          10  GEO-WA2-FN1E-INSTRUC-DIV         PIC X(2).           00027630
          10  FILLER                           PIC X(2).           00027730
          10  GEO-WA2-FN1E-HEALTHAREA          PIC X(4).           00027830
          10  GEO-WA2-FN1E-SANI-REC            PIC X(3).           00027930
          10  GEO-WA2-FN1E-FEATURE-TYPE        PIC X.              00028030
          10  GEO-WA2-FN1E-RESDCP              PIC X.              00028128
          10  GEO-WA2-FN1E-CURVE-FLAG          PIC X.              00028228
          10  GEO-WA2-FN1E-POLICEDIST.                             00028316
             15  GEO-WA2-FN1E-POL-PATR-BORO-CMD PIC X.             00028428
             15  GEO-WA2-FN1E-POL-PRECINCT     PIC X(3).           00028516
          10  GEO-WA2-FN1E-SCHOOLDIST          PIC X(2).           00028616
          10  GEO-WA2-FN1E-ELECTDIST           PIC X(3).           00028716
          10  GEO-WA2-FN1E-ASSEMDIST           PIC X(2).           00028816
          10  GEO-WA2-FN1E-SPLIT-ED-FLAG       PIC X.              00028928
          10  GEO-WA2-FN1E-CONGDIST            PIC X(2).           00029016
          10  GEO-WA2-FN1E-SENATEDIST          PIC X(2).           00029116
          10  GEO-WA2-FN1E-COURTDIST           PIC X(2).           00029216
          10  GEO-WA2-FN1E-COUNCILDIST         PIC X(2).           00029316
          10  FILLER                           PIC X(2).           00029416
          10  GEO-WA2-FN1E-SANIDIST.                               00029516
              15  GEO-WA2-FN1E-SANIDIST-BORO   PIC X.              00029628
              15  GEO-WA2-FN1E-SANIDIST-NUMBER PIC X(2).           00029716
          10  GEO-WA2-FN1E-SANITATION-SUBSEC   PIC X(2).           00029816
** NOTE:10  GEO-WA2-FN1E-FIRESEC ==> FIRE DIVISION **              00029916
          10  GEO-WA2-FN1E-FIRESEC             PIC X(2).           00030016
          10  GEO-WA2-FN1E-FIREBAT             PIC X(2).           00030116
          10  GEO-WA2-FN1E-FIRECO.                                 00030216
              15  GEO-WA2-FN1E-FIRECO-TYPE     PIC X.              00030328
              15  GEO-WA2-FN1E-FIRECO-NUM      PIC X(3).           00030416
          10  GEO-WA2-FN1E-SPECIAL-ADDR-FLAG   PIC X.              00030516
          10  GEO-WA2-FN1E-MARBLE-RIKER-FLAG   PIC X.              00030616
          10  GEO-WA2-FN1E-SPLIT-SCHOOL-FLAG   PIC X.              00030716
          10  GEO-WA2-FN1E-PREFERRED-LGC       PIC X(2).           00030816
          10  GEO-WA2-FN1E-LIONFACECODE        PIC X(4).           00030916
          10  GEO-WA2-FN1E-LIONSEQ             PIC X(5).           00031016
          10  GEO-WA2-FN1E-1990-CENSUSTRACT    PIC X(6).           00031116
          10  FILLER                           PIC X(4).           00031216
          10  GEO-WA2-FN1E-DYN-BLOCK           PIC X(3).           00031312
          10  GEO-WA2-FN1E-XCOORD              PIC X(7).           00031412
          10  GEO-WA2-FN1E-YCOORD              PIC X(7).           00031512
          10  GEO-WA2-FN1E-SEGMENTLENGTH       PIC X(5).           00031612
          10  GEO-WA2-FN1E-SANI-REG            PIC X(5).           00031714
       05  GEO-WA2-FUNCTION5   REDEFINES GEO-WA2-FUNCTION1.        00031812
          10  GEO-WA2-FN5-ADDR-MATCHING-KEY    PIC X(28).          00031912
          10  FILLER                           PIC X(172).         00032012
```

W2COBL COPY File

```
*****************************************************************  00000100
***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2COBL, CONTAINING  ***   00000200
***  THE LAYOUT OF THE OPTIONAL LONG WORK AREA 2 FOR FUNCTIONS *   00000300
***  1/1E AND 3.  THIS WORK AREA SHOULD BE USED ONLY WHEN    ***   00000400
***  FUNCTION 1/1E,3 ARE CALLED WITH THE "LONG" WORK AREA2   ***   00000500
***                                APRIL 3 2001              ***   00000600
*****************************************************************  00000700
***                  LAST MODIFIED    05/19/06               ***   00000812
*****************************************************************  00000904
       05  GEO-WA2-1L-FUNCTION1.                                   00001004
          10  FILLER                           PIC X(21).          00001104
          10  GEO-WA2-1L-CONT-PARITY           PIC X.              00001204
          10  GEO-WA2-1L-LOW-HOUSENUM-INT.                         00001304
             15  GEO-WA2-1L-LOW-HOUSENUM       PIC X(5).           00001404
             15  GEO-WA2-1L-LOW-HOUSENUMSFX    PIC X.              00001504
          10  GEO-WA2-1L-HI-HOUSENUM-INT.                          00001604
             15  GEO-WA2-1L-HI-HOUSENUM        PIC X(5).           00001704
             15  GEO-WA2-1L-HI-HOUSENUMSFX     PIC X.              00001804
          10  GEO-WA2-1L-ALX                   PIC X.              00001912
          10  GEO-WA2-1L-NUM-X-ST-LOW-END      PIC X.              00002000
          10  GEO-WA2-1L-LOW-PBSC              PIC S9(7) COMP-3    00002100
                                               OCCURS 5 TIMES.     00002200
          10  GEO-WA2-1L-NUM-X-ST-HI-END       PIC X.              00002300
          10  GEO-WA2-1L-HI-PBSC               PIC S9(7) COMP-3    00002400
                                               OCCURS 5 TIMES.     00002500
          10  GEO-WA2-1L-COMDIST.                                  00002600
              15  GEO-WA2-1L-COMDIST-BORO      PIC X(1).           00002700
              15  GEO-WA2-1L-COMDIST-NUMBER    PIC X(2).           00002800
          10  GEO-WA2-1L-ZIP                   PIC X(5).           00002900
          10  GEO-WA2-1L-SLA                   PIC X.              00003000
          10  GEO-WA2-1L-HCD                   PIC X(2).           00003100
          10  GEO-WA2-1L-SOS                   PIC X.              00003200
          10  GEO-WA2-1L-CONT-PARITY-IND       PIC X.              00003300
          10  GEO-WA2-1L-2000-CENS-TRCT        PIC X(6).           00003400
          10  GEO-WA2-1L-2000-CENS-BLK         PIC X(4).           00003500
          10  GEO-WA2-1L-INSTRUC-DIV           PIC X(2).           00003604
          10  FILLER                           PIC X(2).           00003704
          10  GEO-WA2-1L-HEALTHAREA            PIC X(4).           00003800
          10  GEO-WA2-1L-SANI-REC              PIC X(3).           00003900
          10  GEO-WA2-1L-FEATURE-TYPE          PIC X(1).           00004103
          10  GEO-WA2-1L-RESDCP                PIC X(1).           00004203
          10  GEO-WA2-1L-CURVE-FLAG            PIC X(1).           00004303
          10  GEO-WA2-1L-POLICEDIST.                               00004403
             15  GEO-WA2-1L-POL-PATR-BORO-CMD  PIC X(1).           00004503
             15  GEO-WA2-1L-POL-PRECINCT       PIC X(3).           00004603
          10  GEO-WA2-1L-SCHOOLDIST            PIC X(2).           00004703
          10  FILLER                           PIC X(15).          00004810
          10  GEO-WA2-1L-SEGMENT-TYPE          PIC X.              00004910
          10  GEO-WA2-1L-SANIDIST.                                 00005003
              15  GEO-WA2-1L-SANIDIST-BORO     PIC X(1).           00005103
              15  GEO-WA2-1L-SANIDIST-NUMBER   PIC X(2).           00005203
          10  GEO-WA2-1L-SANITATION-SUBSEC     PIC X(2).           00005303
  ** NOTE:10  GEO-WA2-1L-FIRESEC ==> FIRE DIVISION **              00005403
          10  GEO-WA2-1L-FIRESEC               PIC X(2).           00005503
          10  GEO-WA2-1L-FIREBAT               PIC X(2).           00005603
          10  GEO-WA2-1L-FIRECO.                                   00005703
              15  GEO-WA2-1L-FIRECO-TYPE       PIC X(1).           00005803
              15  GEO-WA2-1L-FIRECO-NUM        PIC X(3).           00005903
          10  GEO-WA2-1L-SPECIAL-ADDR-FLAG     PIC X(1).           00006003
          10  GEO-WA2-1L-MARBLE-RIKER-FLAG     PIC X(1).           00006103
          10  GEO-WA2-1L-SPLIT-SCHOOL-FLAG     PIC X.              00006203
          10  GEO-WA2-1L-PREFERRED-LGC         PIC X(2).           00006303
          10  GEO-WA2-1L-LIONFACECODE          PIC X(4).           00006403
          10  GEO-WA2-1L-LIONSEQ               PIC X(5).           00006503
          10  GEO-WA2-1L-1990-CENSUSTRACT      PIC X(6).           00006603
```

W2COBL COPY File (continued)

```
           10  FILLER                          PIC X(4).            00006703
           10  GEO-WA2-1L-DYN-BLOCK            PIC X(3).            00006803
           10  GEO-WA2-1L-XCOORD               PIC X(7).            00006903
           10  GEO-WA2-1L-YCOORD               PIC X(7).            00007003
           10  GEO-WA2-1L-SEGMENTLENGTH        PIC X(5).            00007103
           10  GEO-WA2-1L-SANI-REG             PIC X(5).            00007203
           10  GEO-WA2-1L-SEG-ID               PIC X(7).            00007303
           10  GEO-WA2-1L-TRUE-B7SC            PIC X(8).            00007403
           10  GEO-WA2-1L-TRUE-HNI             PIC X(6).            00007507
           10  FILLER                          PIC X(79).           00007607
        05  GEO-WA2-1EL-FUNCTION1E REDEFINES GEO-WA2-1L-FUNCTION1.  00007707
           10  FILLER                          PIC X(21).           00007807
           10  GEO-WA2-1EL-CONT-PARITY         PIC X.               00007907
           10  GEO-WA2-1EL-LOW-HOUSENUM-INT.                        00008007
              15  GEO-WA2-1EL-LOW-HOUSENUM     PIC X(5).            00008107
              15  GEO-WA2-1EL-LOW-HOUSENUMSFX  PIC X.               00008207
           10  GEO-WA2-1EL-HI-HOUSENUM-INT.                         00008307
              15  GEO-WA2-1EL-HI-HOUSENUM      PIC X(5).            00008407
              15  GEO-WA2-1EL-HI-HOUSENUMSFX   PIC X.               00008507
           10  FILLER                          PIC X.               00008607
           10  GEO-WA2-1EL-NUM-X-ST-LOW-END    PIC X.               00008707
           10  GEO-WA2-1EL-LOW-PBSC            PIC S9(7) COMP-3     00008807
                                               OCCURS 5 TIMES.      00008907
           10  GEO-WA2-1EL-NUM-X-ST-HI-END     PIC X.               00009007
           10  GEO-WA2-1EL-HI-PBSC             PIC S9(7) COMP-3     00009107
                                               OCCURS 5 TIMES.      00009207
           10  GEO-WA2-1EL-COMDIST.                                 00009307
               15  GEO-WA2-1EL-COMDIST-BORO    PIC X(1).            00009407
               15  GEO-WA2-1EL-COMDIST-NUMBER  PIC X(2).            00009507
           10  GEO-WA2-1EL-ZIP                 PIC X(5).            00009607
           10  GEO-WA2-1EL-SLA                 PIC X.               00009707
           10  GEO-WA2-1EL-HCD                 PIC X(2).            00009807
           10  GEO-WA2-1EL-SOS                 PIC X.               00009907
           10  GEO-WA2-1EL-CONT-PARITY-IND     PIC X.               00010007
           10  GEO-WA2-1EL-2000-CENS-TRCT      PIC X(6).            00010107
           10  GEO-WA2-1EL-2000-CENS-BLK       PIC X(4).            00010207
           10  GEO-WA2-1EL-INSTRUC-DIV         PIC X(2).            00010307
           10  FILLER                          PIC X(2).            00010407
           10  GEO-WA2-1EL-HEALTHAREA          PIC X(4).            00010507
           10  GEO-WA2-1EL-SANI-REC            PIC X(3).            00010607
           10  GEO-WA2-1EL-FEATURE-TYPE        PIC X(1).            00010707
           10  GEO-WA2-1EL-RESDCP              PIC X(1).            00010800
           10  GEO-WA2-1EL-CURVE-FLAG          PIC X(1).            00010900
           10  GEO-WA2-1EL-POLICEDIST.                              00011000
              15  GEO-WA2-1EL-POL-PATR-BORO-CMD PIC X(1).           00011100
              15  GEO-WA2-1EL-POL-PRECINCT     PIC X(3).            00011200
           10  GEO-WA2-1EL-SCHOOLDIST          PIC X(2).            00011300
           10  GEO-WA2-1EL-ELECTDIST           PIC X(3).            00011400
           10  GEO-WA2-1EL-ASSEMDIST           PIC X(2).            00011500
           10  GEO-WA2-1EL-SPLIT-ED-FLAG       PIC X(1).            00011600
           10  GEO-WA2-1EL-CONGDIST            PIC X(2).            00011700
           10  GEO-WA2-1EL-SENATEDIST          PIC X(2).            00011800
           10  GEO-WA2-1EL-COURTDIST           PIC X(2).            00011900
           10  GEO-WA2-1EL-COUNCILDIST         PIC X(2).            00012000
           10  FILLER                          PIC X(2).            00012100
           10  GEO-WA2-1EL-SANIDIST.                                00012200
               15  GEO-WA2-1EL-SANIDIST-BORO   PIC X(1).            00012300
               15  GEO-WA2-1EL-SANIDIST-NUMBER PIC X(2).            00012400
           10  GEO-WA2-1EL-SANITATION-SUBSEC   PIC X(2).            00012500
** NOTE:10  GEO-WA2-1EL-FIRESEC==> FIRE DIVISION **                 00012600
           10  GEO-WA2-1EL-FIRESEC             PIC X(2).            00012700
           10  GEO-WA2-1EL-FIREBAT             PIC X(2).            00012800
           10  GEO-WA2-1EL-FIRECO.                                  00012900
               15  GEO-WA2-1EL-FIRECO-TYPE     PIC X(1).            00013000
               15  GEO-WA2-1EL-FIRECO-NUM      PIC X(3).            00013100
           10  GEO-WA2-1EL-SPECIAL-ADDR-FLAG   PIC X(1).            00013200
```

W2COBL COPY File (continued)

```
        10  GEO-WA2-1EL-MARBLE-RIKER-FLAG    PIC X(1).            00013300
        10  GEO-WA2-1EL-SPLIT-SCHOOL-FLAG    PIC X.               00013400
        10  GEO-WA2-1EL-PREFERRED-LGC        PIC X(2).            00013500
        10  GEO-WA2-1EL-LIONFACECODE         PIC X(4).            00013600
        10  GEO-WA2-1EL-LIONSEQ              PIC X(5).            00013700
        10  GEO-WA2-1EL-1990-CENSUSTRACT     PIC X(6).            00013800
        10  FILLER                           PIC X(4).            00013900
        10  GEO-WA2-1EL-DYN-BLOCK            PIC X(3).            00014000
        10  GEO-WA2-1EL-XCOORD               PIC X(7).            00014100
        10  GEO-WA2-1EL-YCOORD               PIC X(7).            00014200
        10  GEO-WA2-1EL-SEGMENTLENGTH        PIC X(5).            00014300
        10  GEO-WA2-1EL-SANI-REG             PIC X(5).            00014400
        10  GEO-WA2-1EL-SEG-ID               PIC X(7).            00014500
        10  GEO-WA2-1EL-TRUE-B7SC            PIC X(8).            00014602
        10  GEO-WA2-1EL-TRUE-HNI             PIC X(6).            00014708
        10  FILLER                           PIC X(79).           00014808
     05  GEO-WA2-3L-FUNCTION3 REDEFINES GEO-WA2-1L-FUNCTION1.     00014900
        10  FILLER                           PIC X(21).           00015000
        10  GEO-WA2-3L-DUP-KEY-FLAG          PIC X.               00015100
        10  GEO-WA2-3L-CURVE-FLAG            PIC X.               00015200
        10  GEO-WA2-3L-LOCATION-STATUS       PIC X.               00015300
        10  GEO-WA2-3L-COUNTY-BOUNDARY       PIC X.               00015400
        10  FILLER                           PIC X(4).            00015500
        10  GEO-WA2-3L-PREFERRED-LGC1        PIC X(2).            00015600
        10  GEO-WA2-3L-PREFERRED-LGC2        PIC X(2).            00015700
        10  GEO-WA2-3L-PREFERRED-LGC3        PIC X(2).            00015800
        10  GEO-WA2-3L-NUM-X-ST-LOW-END      PIC X.               00015900
        10  GEO-WA2-3L-LOW-PBSC              PIC S9(7) COMP-3     00016000
                                              OCCURS 5 TIMES.     00016100
        10  GEO-WA2-3L-NUM-X-ST-HI-END       PIC X.               00016200
        10  GEO-WA2-3L-HI-PBSC               PIC S9(7) COMP-3     00016300
                                              OCCURS 5 TIMES.     00016400
        10  GEO-WA2-3L-SLA                   PIC X.               00016500
        10  GEO-WA2-3L-REVERSALFLAG          PIC X.               00016600
        10  GEO-WA2-3L-LEFT-COMDIST.                              00016700
            15  GEO-WA2-3L-LEFT-COMDIST-BORO PIC X(1).            00016800
            15  GEO-WA2-3L-LEFT-COMDIST-NUM  PIC X(2).            00016900
        10  GEO-WA2-3L-RIGHT-COMDIST.                             00017000
            15 GEO-WA2-3L-RIGHT-COMDIST-BORO PIC X(1).            00017100
            15  GEO-WA2-3L-RIGHT-COMDIST-NUM PIC X(2).            00017200
        10  GEO-WA2-3L-LEFT-ZIP              PIC X(5).            00017300
        10  GEO-WA2-3L-RIGHT-ZIP             PIC X(5).            00017400
        10  FILLER                           PIC X(18).           00017500
        10  GEO-WA2-3L-LEFT-HEALTHAREA       PIC X(4).            00017600
        10  GEO-WA2-3L-RIGHT-HEALTHAREA      PIC X(4).            00017700
        10  GEO-WA2-3L-LEFT-INSTRUC-DIV      PIC X(2).            00017804
        10  GEO-WA2-3L-RIGHT-INSTRUC-DIV     PIC X(2).            00017904
        10  GEO-WA2-3L-LEFT-LOW-HOUSENUM     PIC X(7).            00018004
        10  GEO-WA2-3L-LEFT-HI-HOUSENUM      PIC X(7).            00018104
        10  GEO-WA2-3L-RIGHT-LOW-HOUSENUM    PIC X(7).            00018204
        10  GEO-WA2-3L-RIGHT-HI-HOUSENUM     PIC X(7).            00018304
        10  GEO-WA2-3L-CONT-PARITY-IND       PIC X.               00018404
        10  GEO-WA2-3L-LIONFACECODE          PIC X(4).            00018504
        10  GEO-WA2-3L-LIONSEQ               PIC X(5).            00018604
        10  GEO-WA2-3L-GENRECFLAG            PIC X.               00018704
        10  GEO-WA2-3L-SEGMENTLENGTH         PIC S9(5) COMP-3.    00018804
        10  GEO-WA2-3L-SEGMENTSLOPE          PIC X(3).            00018904
        10  GEO-WA2-3L-SEGMENTORIENT         PIC X.               00019004
        10  FILLER                           PIC X(4).            00019104
        10  GEO-WA2-3L-RESDCP                PIC X(2).            00019204
        10  GEO-WA2-3L-DOG-LEG               PIC X.               00019304
        10  GEO-WA2-3L-FEATURE-TYPE          PIC X(1).            00019404
        10  GEO-WA2-3L-LEFT-POLDIST.                              00019504
           15  GEO-WA2-3L-L-POL-PATR-BOR-CMD PIC X(1).            00019604
           15  GEO-WA2-3L-L-POL-PRECINCT     PIC X(3).            00019704
        10  GEO-WA2-3L-RIGHT-POLDIST.                             00019804
```

W2COBL COPY File (continued)

```
              15  GEO-WA2-3L-R-POL-PATR-BOR-CMD PIC X(1).             00019904
              15  GEO-WA2-3L-R-POL-PRECINCT     PIC X(3).             00020004
           10  GEO-WA2-3L-LEFT-SCHLDIST         PIC X(2).             00020104
           10  GEO-WA2-3L-RIGHT-SCHLDIST        PIC X(2).             00020204
           10  GEO-WA2-3L-MARBLE-RIKER-FLAG     PIC X(1).             00020304
           10  GEO-WA2-3L-SEG-ID                PIC X(7).             00020404
           10  GEO-WA2-3L-SEGMENT-TYPE          PIC X.                00020511
*************************************************************         00020704
**   THE PORTION OF THIS WORK AREA ABOVE THIS POINT IS       **       00020804
**   IDENTICAL TO THE STANDARD WORK AREA 2 FOR FUNCTION 3.   **       00020904
**   THE PORTION BELOW THIS POINT IS PRESENT ONLY FOR THE    **       00021004
**   LONG WORK AREA 2 OPTION.                                **       00021104
*************************************************************         00021204
           10  GEO-WA2-3L-L-1990-CENSUSTRACT    PIC X(6) .            00021304
           10  FILLER                           PIC X(4).             00021404
           10  GEO-WA2-3L-L-DYN-BLOCK           PIC X(3).             00021504
           10  GEO-WA2-3L-R-1990-CENSUSTRACT    PIC X(6).             00021604
           10  FILLER                           PIC X(4).             00021704
           10  GEO-WA2-3L-R-DYN-BLOCK           PIC X(3).             00021804
** NOTE:10  GEO-WA2-3L-L-FIRESEC ==> FIRE DIV **                      00021904
** NOTE:10  GEO-WA2-3L-R-FIRESEC ==> FIRE DIV **                      00022004
           10  GEO-WA2-3L-L-FIRESEC             PIC X(2).             00022104
           10  GEO-WA2-3L-L-FIREBAT             PIC X(2).             00022204
           10  GEO-WA2-3L-L-FIRECO.                                   00022304
               15  GEO-WA2-3L-L-FIRECO-TYPE     PIC X(1).             00022404
               15  GEO-WA2-3L-L-FIRECO-NUM      PIC X(3).             00022504
           10  GEO-WA2-3L-R-FIRESEC             PIC X(2).             00022604
           10  GEO-WA2-3L-R-FIREBAT             PIC X(2).             00022704
           10  GEO-WA2-3L-R-FIRECO.                                   00022804
               15  GEO-WA2-3L-R-FIRECO-TYPE     PIC X(1).             00022904
               15  GEO-WA2-3L-R-FIRECO-NUM      PIC X(3).             00023004
           10  GEO-WA2-3L-L-2000-CENS-TRCT      PIC X(6).             00023104
           10  GEO-WA2-3L-L-2000-CENS-BLK       PIC X(4).             00023204
           10  FILLER                           PIC X.                00023304
           10  GEO-WA2-3L-R-2000-CENS-TRCT      PIC X(6).             00023404
           10  GEO-WA2-3L-R-2000-CENS-BLK       PIC X(4).             00023504
           10  FILLER                           PIC X.                00023604
           10  FILLER                           PIC X(36).            00023704
```

W2COB1A COPY File

```
*****************************************************************
**  THIS IS GEOSUPPORT SYSTEM COPY FILE W2COB1A, CONTAINING   **
**  THE LAYOUT OF WORK AREA 2 FOR FUNCTIONS 1A AND BL WHICH   **
**  SHARE A SINGLE WORK AREA 2 LAYOUT.         10/18/96       **
*****************************************************************
 *  NEW FORMAT  *
      05  GEO-WA2-1A-ACCESS-KEY              PIC X(21).
      05  GEO-WA2-1A-CONT-PARITY             PIC X.
      05  GEO-WA2-1A-LOW-HOUSENUM            PIC X(6).
      05  GEO-WA2-1A-ALTKEY-1.
         10 GEO-WA2-1A-ALTKEY-1-BORO         PIC X.
         10 GEO-WA2-1A-ALTKEY-1-TAXBLOCK     PIC X(5).
         10 GEO-WA2-1A-ALTKEY-1-TAXLOT       PIC X(4).
      05  FILLER                             PIC X.
      05  GEO-WA2-1A-SCC                     PIC X.
      05  FILLER                             PIC X.
      05  GEO-WA2-1A-GENERAL-LOT-INFO.
         10 GEO-WA2-1A-RPAD-BLDG-CLASS       PIC X(2).
         10 GEO-WA2-1A-CORNER-CODE           PIC X(2).
         10 GEO-WA2-1A-NUM-OF-STRUCTURES     PIC X(2).
         10 GEO-WA2-1A-NUM-OF-BLOCKFACES     PIC X(2).
         10 GEO-WA2-1A-INTERIOR-FLAG         PIC X.
         10 GEO-WA2-1A-VACANT-FLAG           PIC X.
         10 GEO-WA2-1A-IRREG-FLAG            PIC X.
      05  GEO-WA2-1A-ALT-BORO-FLAG           PIC X.
      05  FILLER                             PIC X.
      05  GEO-WA2-1A-STROLL-KEY              PIC X(13).
      05  GEO-WA2-1A-OVERFLOW-FLAG           PIC X(1).
      05  FILLER-DCP                         PIC X(1).
      05  GEO-WA2-1A-BLDG-ID-NUM             PIC X(7).
      05  GEO-WA2-1A-CONDO-LOT-FLAG          PIC X.
      05  GEO-WA2-1A-RPAD-COND-NUM           PIC X(4).
      05  GEO-WA2-1A-CONDO-LOW-BBL           PIC X(10).
      05  FILLER                             PIC X.
      05  GEO-WA2-1A-CONDO-BILLING-BBL       PIC X(10).
      05  FILLER                             PIC X.
      05  GEO-WA2-1A-CONDO-BILL-BBL-SCC      PIC X.
      05  GEO-WA2-1A-CONDO-HIGH-BBL          PIC X(10).
      05  FILLER                             PIC X.
      05  GEO-WA2-1A-SANBORN-BVOLPAGE.
          10  GEO-WA2-1A-SANBORN-BORO        PIC X(1).
          10  GEO-WA2-1A-SANBORN-VOL-PAGE.
              15  GEO-WA2-1A-SANBORN-VOL-NUM   PIC X(3).
              15  GEO-WA2-1A-SANBORN-PAGE-NUM  PIC X(4).
      05  GEO-WA2-1A-COMMERC-DIST            PIC X(5).
      05  GEO-WA2-1A-CO-OP-NBR               PIC X(4).
      05  FILLER                             PIC X(4).
      05  GEO-WA2-1A-TOT-NBR-BLDG            PIC X(4).
      05  GEO-WA2-1A-DOF-MAP-BOROUGH         PIC X.
      05  GEO-WA2-1A-TAX-MAP-NBR             PIC X(4).
      05  FILLER-FOR-TAX-MAP-PAGE            PIC X(4).
      05  GEO-WA2-1A-X-COORD                 PIC X(7).
      05  GEO-WA2-1A-Y-COORD                 PIC X(7).
      05  FILLER                             PIC X(10).
      05  FILLER-FOR-LGC-LIST                PIC X(8).
      05  GEO-WA2-1A-NUM-OF-ADDR-FOR-LOT     PIC X(2).
      05  GEO-WA2-1A-LIST-OF-ADDRESSES       OCCURS 21 TIMES.
         10  GEO-WA2-1A-LIST-LOW-HOUSENUM    PIC X(6).
         10  FILLER                          PIC X(3).
         10  GEO-WA2-1A-LIST-HI-HOUSENUM     PIC X(6).
         10  FILLER                          PIC X(3).
         10  GEO-WA2-1A-LIST-STREETCODE      PIC X(8).
         10  GEO-WA2-1A-LIST-BIN             PIC X(7).
         10  GEO-WA2-1A-ADDR-TYPE            PIC X.
         10  FILLER                          PIC X.
         10  GEO-WA2-1A-LIST-SOS             PIC X.
```

W2COB1AL COPY File

```
*****************************************************************  00000100
**   THIS IS GEOSUPPORT SYSTEM COPY FILE W2COB1AL, CONTAINING   **  00000200
**   THE LAYOUT OF LONG WORK AREA 2 FOR FUNCTIONS 1A AND BL     **  00000300
**   WHICH SHARE A SINGLE WORK AREA 2 LAYOUT.       11/06/97    **  00000400
*****************************************************************  00000500
 *   1A/BL LONG WORK AREA 2 *                                       00000600
         05  GEO-WA2-1AL-ACCESS-KEY             PIC X(21).          00000700
         05  GEO-WA2-1AL-CONT-PARITY            PIC X.              00000800
         05  GEO-WA2-1AL-LOW-HOUSENUM           PIC X(6).           00000900
         05  GEO-WA2-1AL-ALTKEY-1.                                  00001000
            10 GEO-WA2-1AL-ALTKEY-1-BORO        PIC X.              00001100
            10 GEO-WA2-1AL-ALTKEY-1-TAXBLOCK    PIC X(5).           00001200
            10 GEO-WA2-1AL-ALTKEY-1-TAXLOT      PIC X(4).           00001300
         05  FILLER                            PIC X.               00001400
         05  GEO-WA2-1AL-SCC                    PIC X.              00001500
         05  FILLER                            PIC X.               00001600
         05  GEO-WA2-1AL-GENERAL-LOT-INFO.                          00001700
            10 GEO-WA2-1AL-RPAD-BLDG-CLASS      PIC X(2).           00001800
            10 GEO-WA2-1AL-CORNER-CODE          PIC X(2).           00001900
            10 GEO-WA2-1AL-NUM-OF-STRUCTURES    PIC X(2).           00002000
            10 GEO-WA2-1AL-NUM-OF-BLOCKFACES    PIC X(2).           00002100
            10 GEO-WA2-1AL-INTERIOR-FLAG        PIC X.              00002200
            10 GEO-WA2-1AL-VACANT-FLAG          PIC X.              00002300
            10 GEO-WA2-1AL-IRREG-FLAG           PIC X.              00002400
         05  GEO-WA2-1AL-ALT-BORO-FLAG          PIC X.              00002500
         05  FILLER                            PIC X.               00002600
         05  GEO-WA2-1AL-STROLL-KEY             PIC X(13).          00002700
         05  FILLER                            PIC X(1).            00002800
         05  FILLER-DCP                        PIC X(1).            00002900
         05  GEO-WA2-1AL-BLDG-ID-NUM            PIC X(7).           00003000
         05  GEO-WA2-1AL-CONDO-LOT-FLAG         PIC X.              00003100
         05  GEO-WA2-1AL-RPAD-COND-NUM          PIC X(4).           00003200
         05  GEO-WA2-1AL-CONDO-LOW-BBL          PIC X(10).          00003300
         05  FILLER                            PIC X.               00003400
         05  GEO-WA2-1AL-CONDO-BILLING-BBL      PIC X(10).          00003500
         05  FILLER                            PIC X.               00003600
         05  GEO-WA2-1AL-CONDO-BILL-BBL-SCC     PIC X.              00003700
         05  GEO-WA2-1AL-CONDO-HIGH-BBL         PIC X(10).          00003800
         05  FILLER                            PIC X.               00003900
         05  GEO-WA2-1AL-SANBORN-BVOLPAGE.                          00004000
             10  GEO-WA2-1AL-SANBORN-BORO       PIC X(1).           00004100
             10  GEO-WA2-1AL-SANBORN-VOL-PAGE.                      00004200
                 15  GEO-WA2-1AL-SANBORN-VOL-NUM   PIC X(3).        00004300
                 15  GEO-WA2-1AL-SANBORN-PAGE-NUM  PIC X(4).        00004400
         05  GEO-WA2-1AL-COMMERC-DIST           PIC X(5).           00004500
         05  GEO-WA2-1AL-CO-OP-NBR              PIC X(4).           00004604
         05  FILLER                             PIC X(4).           00004705
         05  GEO-WA2-1AL-TOT-NBR-BLDG           PIC X(4).           00004804
         05  GEO-WA2-1AL-DOF-MAP-BORO           PIC X.              00004907
         05  GEO-WA2-1AL-DOF-MAP-SECVOL         PIC X(4).           00005007
  *******    GEO-WA2-1AL-DOF-MAP-PAGE  NOT IMPLEMENTED              00005107
         05  GEO-WA2-1AL-DOF-MAP-PAGE           PIC X(4).           00005207
         05  GEO-WA2-1AL-X-COORD                PIC X(7).           00005308
         05  GEO-WA2-1AL-Y-COORD                PIC X(7).           00005408
         05  FILLER                             PIC X(16).          00005508
         05  GEO-WA2-1AL-NUM-OF-BINS            PIC X(4).           00005608
         05  GEO-WA2-1AL-BINS                   PIC X(7)            00005708
                                               OCCURS 2500 TIMES.   00006004
```

W2COB3S COPY File

```
*****************************************************************
***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2COB3S, CONTAINING    **
***  THE LAYOUT OF WORK AREA 2 FOR FUNCTION 3S.         9/22/93 **
*****************************************************************
       05  GEO-WA2-3S-ACCESS-KEY                PIC X(21).
       05  GEO-WA2-3S-NUM-OF-INTERSECTS         PIC X(3).
       05  GEO-WA2-3S-LIST-OFINTERSECTS         OCCURS 350 TIMES.
          10 GEO-WA2-3S-SMALLEST-PBSC           PIC S9(7) COMP-3.
          10 GEO-WA2-3S-2ND-SMALLEST-PBSC       PIC S9(7) COMP-3.
          10 GEO-WA2-3S-DISTANCE                PIC S9(5) COMP-3.
          10 GEO-WA2-3S-GAP-FLAG                PIC X.
```

ASSEMBLER COPY Files (MSW)

W1BAL COPY File

```
*/*******************************************************************/
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE W1BAL,    ***/
*/*****  CONTAINING THE LAYOUT OF WORK AREA 1.                     ***/
*/*******************************************************************/
*/*****  Last Updated - 7 February 2006                            ***/
*/*******************************************************************/
W1BAL    DS   0H
*/****************************
*/*****  INPUT FIELDS  *******
*/****************************
W1IFUNC  DS    0CL2      FUNCTION CODE
W1IFUNC1 DS     CL1      FUNCTION CODE, BYTE 1
W1IFUNC2 DS     CL1      FUNCTION CODE, BYTE 2
W1IBORO  DS    0CL1
W1IBORO1 DS     CL1      BORO CODE  (1=MN;2=BX;3=BK;4=QN;5=SI)
W1IHSE#  DS    CL12      UNFORMATED HSNUM FOR FUNCTION: 1; 1A; 1E.
W1IHSE#P DS    CL6       HOUSE NUM (INTERNAL FORMAT FOR FUNC D)
W1ISTRT1 DS    CL32      STREET NAME 1
W1ISTRT2 DS    CL32      STREET NAME 2
W1ISTRT3 DS    CL32      STREET NAME 3
W1ICOMP  DS    CL1       COMPASS DIRECTION (TYPES 2, 3C & 3S)
W1ICOMP2 DS    CL1       COMPASS DIRECTION (TYPE 3S)
W1ICDE1  DS    PL4       PB5SC FOR STREET 1
W1ICDE2  DS    PL4       PB5SC FOR STREET 2
W1ICDE3  DS    PL4       PB5SC FOR STREET 3
W1IRBRQS DS    CL1       ROADBED REQUEST SWITCH
W1IBORO2 DS    CL1       BORO CODE OF CROSS ST. 1
W1IBORO3 DS    CL1       BORO CODE OF CROSS ST. 2
W1ISNL   DS    CL2       LENGTH STREET NAME IS TO BE NORMALIZED TO
W1I10SC1 DS    CL11      BORO + 10 BYTE STREET CODE FOR CROSS STREET 1
W1I10SC2 DS    CL11      BORO + 10 BYTE STREET CODE FOR CROSS STREET 2
W1I10SC3 DS    CL11      BORO + 10 BYTE STREET CODE FOR CROSS STREET 3
W1ICONDO DS    CL5       CONDO UNIT ID NUMBER - NOT IMPLEMENTED
W1IBBL   DS    0CL10     BORO,BLOCK,LOT FOR "BL" FUNCTION
W1IBLBOR DS     CL1      BORO FOR FUNCTION "BL"
W1IBLOCK DS     CL5      TAX BLOCK - FOR  FUNCTION "BL"
W1ILOT   DS     CL4      TAX LOT   - FOR  FUNCTION "BL"
         DS    CL1       FILLER
W1IBIN   DS    CL7       BUILDING ID NUMBER
W1ICMPCT DS    CL1       'C' IF STREET NAMES ARE TO BE COMPACTED
W1ILONG3 DS    CL1       'L' IF LONG WORKAREA 2 FOR FUNC 3 DESIRED
W1ILHSE  DS    CL12      UNFORMATED LOW HSNUM FOR FUNCTION: 1; 1A; 1E.
W1ILHSEP DS    CL6       LOW HOUSE NUM (INTERNAL FORMAT FOR FUNC D)
W1INIBMF DS    CL1       NON-IBM MAIN FRAMME FLAG
W1I1ABLV DS    CL1       Set to "S" for St'd Func. 1A & BL WORKAREA
*                        Set to "L" or " " for Legacy 1A & BL Workarea
W1IXSTF  DS    CL1       CROSS STREET NAME FLAG
         DS    CL4       FILLER
*/*****************************
*/*****  OUTPUT FIELDS  *******
*/*****************************
W1OLHSE  DS    CL12      LOW HOUSE NUMBER IN DISPLAY FORMAT
W1OBORO  DS    CL9       BORO NAME
W1OSTRT1 DS    CL32      STREET 1 NAME, NORMALIZED
W1OSTRT2 DS    CL32      STREET 2 NAME, NORMALIZED
W1OSTRT3 DS    CL32      STREET 3 NAME, NORMALIZED
W1OHSE#  DS    CL12      HOUSE NUMBER, NORMALIZED, DISPLAY FORMAT
W1OHSE#P DS    CL6       HOUSE NUMBER (INTERNAL FORMAT)
         DS    CL7       FILLER
W1OPB51K DS    0PL4      Packed Borough and Street Code 1
W1OBOR1K DS    XL1       Packed unsigned Borough Code
W1OCDE1K DS    PL3       STREET CODE 1 (KEY)
         DS    PL2       Filler
W1OPB52K DS    0PL4      Packed Borough and Street Code 2
```

W1BAL COPY File (continued)

```
W1OBOR2K DS    XL1       Packed unsigned Borough Code                     00006000
W1OCDE2K DS    PL3       STREET CODE 2 (KEY)                              00006100
         DS    PL2       Filler                                           00006200
W1OPB53K DS    0PL4      Packed Borough and Street Code 3                 00006300
W1OBOR3K DS    XL1       Packed unsigned Borough Code                     00006400
W1OCDE3K DS    PL3       STREET CODE 3 (KEY)                              00006500
W1OATTR  DS    CL3       Attribute Bytes - Internal Use Only              00006600
W1BROWSE DS    CL40      10 PB5SC'S FOR  FUNCTION: BB; BF.                00006700
W1O10SC1 DS    CL11      BORO + 10 BYTE STREET CODE FOR CROSS STREET 1    00006800
W1O10SC2 DS    CL11      BORO + 10 BYTE STREET CODE FOR CROSS STREET 2    00006900
W1O10SC3 DS    CL11      BORO + 10 BYTE STREET CODE FOR CROSS STREET 2    00007000
W1OCONDO DS    CL5       CONDO UNIT ID NUMBER - NOT IMPLEMENTED           00007100
W1OBBL   DS    0CL10     OUTPUT BORO,BLOCK,LOT FOR FUNCTION "BL"          00007200
W1OBLBOR DS     CL1      BORO FOR FUNCTION "BL"                           00007300
W1OBLOCK DS     CL5      TAX BLOCK - FOR FUNCTION "BL"                    00007400
W1OLOT   DS     CL4      TAX LOT   - FOR FUNCTION "BL"                    00007500
         DS    CL1     FILLER                                             00007510
W1OBIN   DS    CL7     BUILDING IDENTIFICATION Number                     00007600
W1OINTU1 DS    CL1       INTERNAL USE ONLY - ATTR BYTE
W1OREASN DS    CL1       REASON CODE                                      00007900
W1OINTR0 DS    CL1       INTERNAL USE ONLY - RETURN CODE                  00008000
W1OINTRC DS    CL1       INTERNAL USE ONLY - RETURN CODE                  00008100
W1ORC    DS    0CL2      RETURN CODE                                      00008200
W1ORC1   DS     CL1      RETURN CODE, BYTE 1                              00008300
W1ORC2   DS     CL1      RETURN CODE, BYTE 2                              00008400
W1OERROR DS    CL80      ERROR MESSAGE                                    00008500
W1O#SIM  DS    PL2       NUMBER OF SIMILAR NAMES                          00008600
W1ONAMES DS  10CL32      UP TO 10 SIMILAR NAMES                           00008700
W1END    EQU   *                                                          00008800
W1LENGTH EQU   W1END-W1BAL    LENGTH OF W1BAL                             00008900
```

W2BAL COPY File

```
*/*******************************************************************/ 00010000
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE W2BAL,    ***/ 00020000
*/*****  CONTAINING THE LAYOUT OF WORK AREA 2 FOR FUNCTIONS        ***/ 00030000
*/*****  1, 1E, 2, 2C, 3, 3C. PLEASE NOTE THAT FUNCTIONS 2 AND 2C  ***/ 00040000
*/*****  SHARE A SINGLE WORK AREA 2 LAYOUT.                        ***/ 00050000
*/*******************************************************************/ 00060000
*/*****              LAST MODIFIED  7 FEBRUARY 2006                ***/ 00070027
*/*******************************************************************/ 00080012
W2BAL    DS   0H                                                        00090000
W2ACCKEY DS    CL21         ACCESS KEY                                  00100000
W2LAYOUT DS   0CL179                                                    00110002
W2F1CPAR DS     CL1         CONTINUOUS PARITY INDICATOR                 00120002
W2F1LHNI DS    0CL6         LOW HOUSE NUMBER                            00130000
W2F1HSEL DS     CL5         LOW HOUSE NUMBER ON BLOCK FACE              00140000
W2F1SFXL DS     CL1         LOW HOUSE NUMBER SUFFIX                     00150013
W2F1HHNI DS    0CL6         HIGH HOUSE NUMBER                           00160000
W2F1HSEH DS     CL5         HIGH HOUSE NUMBER ON BLOCK FACE             00170000
W2F1SFXH DS     CL1         HI HOUSE NUMBER SUFFIX                      00180013
W2F1ALX  DS    CL1          A=ALLEYS INTERSECT SEGMENT                  00190025
*                           X=CROSS STREETS MODIFIED                    00191025
W2F1#STL DS    CL1          NUMBER OF CROSS STREETS AT LOW END          00200000
W2F1CDEL DS    CL20         UP TO FIVE PB5SC'S FOR LOW END              00210000
W2F1#STH DS    CL1          NUMBER OF CROSS STREETS AT HIGH END         00220000
W2F1CDEH DS    CL20         UP TO FIVE PB5SC'S FOR HIGH END             00230000
W2F1CD   DS    0CL3         COMMUNITY DISTRICT                          00240000
W2F1CDB  DS     CL1         COMMUNITY DISTRICT BORO                     00250000
W2F1CDN  DS     CL2         COMMUNITY DISTRICT NUMBER                   00260000
W2F1ZIP  DS    CL5          ZIP CODE                                    00270000
W2F1SLA  DS    CL1          STREET LIGHT AREA                           00280000
W2F1HCD  DS    CL2          HEALTH CODE DISTRICT                        00290000
W2F1SOS  DS    CL1          SIDE OF STREET INDICATOR                    00300000
W2F1PAR  DS    CL1          CONTINUOUS PARITY INDICATOR                 00310000
W2F1CT00 DS    CL6          2000 CENSUS TRACT                           00320015
W2F1CB00 DS    CL4          2000 CENSUS BLOCK                           00350015
W2F1INSD DS    CL2          INSTRUCTIONAL DIVISION                      00351021
         DS    CL2          FILLER                                      00352021
W2F1HA   DS    CL4          HEALTH AREA                                 00360000
W2F1SREC DS    CL3          SANITATION RECYCLE PICK-UP                  00370007
W2F1FEAT DS    CL1          FEATURE TYPE CODE                           00380023
RES1     DS    CL1          RESERVED FOR DCP/GSS USE                    00400000
W2F1CURV DS    CL1          CURVE FLAG                                  00410014
W2F1POL  DS    0CL4         POLICE DISTRICT                             00420000
W2F1PBC  DS     CL1         POLICE PATROL BORO COMMAND                  00430000
W2F1POP  DS     CL3         POLICE PRECINCT                             00440000
W2F1SCH  DS    CL2          SCHOOL DISTRICT                             00450000
         DS    CL15        RESERVED FOR POLITICAL INFORMATION           00460026
W2F1STC  DS    CL1          SEGMENT TYPE CODE                           00461026
W2F1SAND DS    CL3          SANITATION DISTRICT                         00470002
W2F1SANT DS    CL2          SANITATION DEPT SUBSECTION                  00480000
W2F1FS   DS    CL2          FIRE DIVISION                               00490001
W2F1FB   DS    CL2          FIRE BATTALION                              00500000
W2F1FC   DS    0CL4         FIRE COMPANY                                00510000
W2F1FCT  DS     CL1         FIRE COMPANY TYPE                           00520000
W2F1FCN  DS     CL3         FIRE COMPANY NUMBER                         00530000
W2F1SPAD DS    CL1          SPECIAL ADDRESS FLAG                        00540000
W2F1MHRI DS    CL1          MARBLE HILL/RIKERS ISLAND FLAG              00550000
W2F1SSCH DS    CL1          SPLIT SCHOOL DISTRICT FLAG                  00560000
W2F1LGC  DS    CL2          LOGICAL GROUP CODE (PREFERRED)              00570000
W2F1FACE DS    CL4          LION FACE CODE                              00580000
W2F1SEQ  DS    CL5          LION SEQUENCE NUMBER                        00590000
W2F1CT90 DS    CL6          1990 CENSUS TRACT                           00600000
         DS    CL4          FILLER                                      00610015
W2F1CPB  DS    CL3          DYNAMIC  BLOCK                              00640000
W2F1XCOR DS    CL7          X COORDINATE                                00650000
W2F1YCOR DS    CL7          Y COORDINATE                                00660000
```

W2BAL COPY File (continued)

```
W2F1SEGL DS    CL5           SEGMENT LEGNTH                               00670000
W2F1SREG DS    CL5           SANITATION REGULAR PICK-UP                   00680012
*                                                                         00690000
********************************************************************      00700000
         ORG   W2F1SCH+2     PATCH FOR FUNCTION 1E FIELDS                 00710000
********************************************************************      00720000
*                                                                         00730000
W2F1EED  DS    CL3           ELECTION DISTRICT                            00740000
W2F1EAD  DS    CL2           ASSEMBLY DISTRICT                            00750000
W2F1ESED DS    CL1           SPLIT E.D. FLAG                              00760000
W2F1ECON DS    CL2           CONGRESSIONAL DISTRICT                       00770000
W2F1ESEN DS    CL2           SENATORIAL DISTRICT                          00780000
W2F1ECIV DS    CL2           CIVIL COURT DISTRICT                         00790000
W2F1ECOU DS    CL2           CITY COUNCIL DISTRICT                        00800000
         DS    CL18                                                       00810000
W2F1ELGC DS    CL2           LOGICAL GROUP CODE (PREFERRED)               00820000
*                                                                         00830000
********************************************************************      00840000
         ORG   W2LAYOUT      RESET LOCATION COUNTER FOR FUNCTION 2        00850000
********************************************************************      00860000
*                                                                         00870000
W2F2DUPI DS    CL1           DUPLICATE INTERSECT FLAG                     00880000
         DS    CL9           FILLER                                       00890000
W2F2LGC1 DS    CL2           STREET 1 PREFERRED LGC                       00900013
W2F2LGC2 DS    CL2           STREET 2 PREFERRED LGC                       00910013
W2F2#INT DS    CL1           NUMBER OF INTERSECTING STREETS               00920000
W2F2CODE DS    CL20          INTERSECTING PB5SC'S                         00930000
W2F2CDIR DS    CL1           COMPASS DIRECTION OF TWO LOWEST STREETS      00940011
W2F2LEVC DS    CL10          LEVEL CODES ASSOCIATED WITH CROSS STREETS    00941022
W2F2INSD DS    CL2           INSTRUCTIONAL DIVISION                       00950022
W2F2FS   DS    CL2           FIRE DIVISION                                00960003
W2F2FB   DS    CL2           FIRE BATTALION                               00970000
W2F2FC   DS    0CL4          FIRE COMPANY                                 00980000
W2F2FCT  DS     CL1          FIRE COMPANY TYPE                            00990000
W2F2FCN  DS     CL3          FIRE COMPANY NUMBER                          01000000
W2F2CD   DS    0CL3          COMMUNITY DISTRICT                           01010000
W2F2CDB  DS     CL1          COMMUNITY DISTRICT BORO                      01020000
W2F2CDN  DS     CL2          COMMUNITY DISTRICT NUMBER                    01030000
W2F2ZIP  DS    CL5           ZIP CODE                                     01040000
W2F2SLA  DS    CL1           STREET LIGHT AREA                            01050000
W2F2CT00 DS    CL6           2000 CENSUS TRACT                            01060015
         DS    CL3           FILLER                                       01080000
W2F2HA   DS    CL4           HEALTH AREA                                  01090000
         DS    CL9           FILLER                                       01100010
W2F2NDNB DS    CL7           LION NODE NUMBER                             01110010
W2F2XCOR DS    CL7           X COORDINATE                                 01120000
W2F2YCOR DS    CL7           Y COORDINATE                                 01130000
         DS    CL4           FILLER                                       01150013
W2F2POL  DS    0CL4          POLICE DISTRICT                              01160000
W2F2PBC  DS     CL1          POLICE PATROL BORO COMMAND                   01170000
W2F2POP  DS     CL3          POLICE PRECINCT                              01180000
W2F2SCH  DS    CL2           SCHOOL DISTRICT                              01190000
W2F2MHRI DS    CL1           MARBLE HILL/RIKERS ISLAND FLAG               01200000
W2F2CT90 DS    CL6           1990 CENSUS TRACT                            01210000
W2F2SVP1 DS    0CL8          FIRST SANBORN BOROUGH, PAGE, VOLUME          01220003
W2F2SB1  DS    CL1           FIRST SANBORN BOROUGH CODE                   01230003
W2F2SP1  DS    CL3           FIRST SANBORN PAGE                           01240003
W2F2SV1  DS    CL4           FIRST SANBORN VOLUME                         01250003
W2F2SVP2 DS    0CL8          SECOND SANBORN BOROUGH, PAGE, VOLUME         01260003
W2F2SB2  DS    CL1           SECOND SANBORN BOROUGH CODE                  01270003
W2F2SP2  DS    CL3           SECOND SANBORN PAGE                          01280003
W2F2SV2  DS    CL4           SECOND SANBORN VOLUME                        01290003
         DS    CL38          FILLER                                       01300003
*                                                                         01310000
********************************************************************      01320000
         ORG   W2LAYOUT      RESET LOCATION COUNTER FOR FUNCTION 3        01330000
```

W2BAL COPY File (continued)

```
*********************************************************************    01340000
*                                                                        01350000
W2F3DUPF DS    CL1           DUPLICATE KEY FLAG                          01360013
W2F3CURV DS    CL1           CURVE FLAG                                  01361014
W2F3LST  DS    CL1           LOCATIONAL STATUS                           01362018
W2F3CBI  DS    CL1           COUNTY BOUNDARY INDICATOR                   01363017
         DS    CL4                                                       01370017
W2F3LGC1 DS    CL2           STREET 1 PREFERRED LGC                      01380013
W2F3LGC2 DS    CL2           STREET 2 PREFERRED LGC                      01390013
W2F3LGC3 DS    CL2           STREET 3 PREFERRED LGC                      01400013
W2F3#STL DS    CL1           NUMBER OF CROSS STREETS AT LOW END          01410000
W2F3CDEL DS    CL20          CROSS STREET PB5SC'S AT LOW END             01420000
W2F3#STH DS    CL1           NUMBER OF CROSS STREETS AT HIGH END         01430000
W2F3CDEH DS    CL20          CROSS STREET PB5SC'S AT HIGH END            01440000
W2F3SLA  DS    CL1           STREET LIGHT AREA                           01450000
W2F3REVF DS    CL1           REVERSAL FLAG                               01460000
W2F3CDL  DS    0CL3          LEFT COMMUNITY DISTRICT                     01470000
W2F3CDBL DS    CL1           LEFT COMMUNITY DISTRICT BORO                01480006
W2F3CDNL DS    CL2           LEFT COMMUNITY DISTRICT NUMBER              01490006
W2F3CDR  DS    0CL3          RIGHT COMMUNITY DISTRICT                    01500000
W2F3CDBR DS     CL1          RIGHT COMMUNITY DISTRICT BORO               01510006
W2F3CDNR DS     CL2          RIGHT COMMUNITY DISTRICT NUMBER             01520006
W2F3ZIPL DS    CL5           LEFT ZIP CODE                               01530000
W2F3ZIPR DS    CL5           RIGHT ZIP CODE                              01540000
         DS    CL18          FILLER - FORMER 1980 CENSUS GEOGRAPHY       01541015
W2F3HAL  DS    CL4           LEFT HEALTH AREA                            01610000
W2F3HAR  DS    CL4           RIGHT HEALTH AREA                           01620000
W2F3INSL DS    CL2           LEFT INSTRUCTIONAL DIVISION                 01630022
W2F3INSR DS    CL2           RIGHT INSTRUCTIONAL DIVISION                01631022
W2F3LO#L DS    CL7           LEFT LOW HOUSE NUMBER                       01640000
W2F3HI#L DS    CL7           LEFT HIGH HOUSE NUMBER                      01650000
W2F3LO#R DS    CL7           RIGHT LOW HOUSE NUMBER                      01660000
W2F3HI#R DS    CL7           RIGHT HIGH HOUSE NUMBER                     01670000
W2F3PAR  DS    CL1           CONTINUOUS PARITY INDICATOR                 01680000
W2F3FACE DS    CL4           LION FACE CODE                              01690000
W2F3SEQ  DS    CL5           LION SEQUENCE NUMBER                        01700000
W2F3GEN  DS    CL1           GENERATED RECORD FLAG                       01710000
W2F3SEGL DS    PL3           SEGMENT LENGTH IN FEET                      01720000
W2F3SLOP DS    CL3           SEGMENT SLOPE IN DEGREES                    01730000
W2F3ORNT DS    CL1           SEGMENT ORIENTATION                         01740000
         DS    CL4           FILLER                                      01750013
RES2     DS    CL2           RESERVED FOR DCP/GSS USE                    01770000
W2F3DGLG DS    CL1           DOG LEG FLAG                                01771015
W2F3FEAT DS    CL1           FEATURE TYPE CODE                           01780024
W2F3POLL DS    0CL4          LEFT POLICE DISTRICT                        01790000
W2F3PBCL DS     CL1          LEFT POLICE PATROL BORO COMMAND             01800000
W2F3POPL DS     CL3          LEFT POLICE PRECINCT                        01810000
W2F3POLR DS    0CL4          RIGHT POLICE DISTRICT                       01820000
W2F3PBCR DS     CL1          RIGHT POLICE PATROL BORO COMMAND            01830000
W2F3POPR DS     CL3          RIGHT POLICE PRECINCT                       01840000
W2F3SCHL DS    CL2           LEFT SCHOOL DISTRICT                        01850000
W2F3SCHR DS    CL2           RIGHT SCHOOL DISTRICT                       01860000
W2F3MHRI DS    CL1           MARBLE HILL/RIKERS ISLAND FLAG              01870000
W2F3SEGT DS    CL7           SEGMENT IDENTIFIER                          01871015
W2F3STC  DS    CL1           SEGMENT TYPE CODE                           01880026
*                                                                        01890000
*********************************************************************    01900000
         ORG   W2LAYOUT      RESET LOCATION COUNTER FOR FUNCTION 3C      01910000
*********************************************************************    01920000
*                                                                        01930000
W23CCURV DS    CL1           CURVE FLAG                                  01931014
W23CSTC  DS    CL1           SEGMENT TYPE CODE                           01932026
W23CLST  DS    CL1           LOCATIONAL STATUS                           01933018
W23CCBI  DS    CL1           COUNTY BOUNDARY INDICATOR                   01934017
         DS    CL4           FILLER                                      01940017
W23CLGC1 DS    CL2           STREET 1 PREFERRED LGC                      01950013
```

W2BAL COPY File (continued)

```
W23CLGC2 DS    CL2            STREET 2 PREFERRED LGC                    01960013
W23CLGC3 DS    CL2            STREET 3 PREFERRED LGC                    01970013
W23C#STL DS    CL1            NUMBER OF CROSS STREETS AT LOW END        01980000
W23CCDEL DS    CL20           UP TO FIVE PB5SC'S FOR LOW END            01990000
W23C#STH DS    CL1            NUMBER OF CROSS STREETS AT HIGH END       02000000
W23CCDEH DS    CL20           UP TO FIVE PB5SC'S FOR HIGH END           02010000
W23CCD   DS    0CL3           COMMUNITY DISTRICT                        02020000
W23CCDB  DS     CL1           COMMUNITY DISTRICT BORO                   02030000
W23CCDN  DS     CL2           COMMUNITY DISTRICT NUMBER                 02040000
W23CZIP  DS    CL5            ZIP CODE                                  02050000
W23CSLA  DS    CL1            STREET LIGHT AREA                         02060000
         DS    CL7            FILLER                                    02080000
W23CCT00 DS    CL6            2000 CENSUS TRACT                         02100015
W23CCB00 DS    CL4            2000 CENSUS BLOCK                         02110015
         DS    CL1            POSSIBLE CENSUS BLOCK SUFFIX              02111015
W23CHA   DS    CL4            HEALTH AREA                               02120000
W23CREVF DS    CL1            CROSS STREET REVERSAL FLAG                02130005
W23CSOS  DS    CL1            SIDE OF STREET INDICATOR                  02140005
W23CFS   DS    CL2            FIRE DIVISION                             02150003
W23CFB   DS    CL2            FIRE BATTALION                            02160000
W23CFC   DS    0CL4           FIRE COMPANY                              02170000
W23CFCT  DS     CL1           FIRE COMPANY TYPE                         02180000
W23CFCN  DS     CL3           FIRE COMPANY NUMBER                       02190000
W23CSEGT DS    CL7            SEGMENT IDENTIFIER                        02200015
W23CHSEL DS    CL7            LOW HOUSE NUMBER                          02210000
W23CHSEH DS    CL7            HIGH HOUSE NUMBER                         02220000
W23CHS2L DS    CL7            2ND LOW HSE # - USED IF ODD & EVEN RANGES  02230000
W23CHS2H DS    CL7            2ND HI HSE #    ARE ON SAME SIDE OF STREET 02240000
W23CPAR  DS    CL1            CONTINUOUS PARITY INDICATOR               02250000
W23CFACE DS    CL4            LION FACE CODE                            02260000
W23CSEQ  DS    CL5            LION SEQUENCE NUMBER                      02270000
W23CGEN  DS    CL1            GENERATED RECORD FLAG                     02280000
W23CSEGL DS    PL3            SEGMENT LENGTH IN FEET                    02290000
W23CSLOP DS    CL3            SEGMENT SLOPE IN DEGREES                  02300000
W23CORNT DS    CL1            SEGMENT ORIENTATION                       02310000
W23CINSD DS    CL2            INSTRUCTIONAL DIVISION                    02320022
RES3     DS    CL1            RESERVED FOR DCP/GSS USE                  02330000
W23CFEAT DS    CL1            FEATURE TYPE CODE                         02340024
W23CPOL  DS    0CL4           POLICE DISTRICT                           02350000
W23CPBC  DS     CL1           POLICE PATROL BORO COMMAND                02360000
W23CPOP  DS     CL3           POLICE PRECINCT                           02370000
W23CSCH  DS    CL2            SCHOOL DISTRICT                           02380000
W23CMHRI DS    CL1            MARBLE HILL/RIKERS ISLAND FLAG            02390000
W23CCT90 DS    CL6            1990 CENSUS TRACT                         02400000
         DS    CL4            FILLER                                    02410015
W23CCPB  DS    CL3            DYNAMIC  BLOCK                            02440000
         DS    CL5            FILLER                                    02450000
*                                                                       02460000
*********************************************************************** 02470000
         ORG   W2BAL          RESET LOCATION COUNTER FOR FUNCTION 5     02480000
*********************************************************************** 02490000
*                                                                       02500000
W2F5AMK  DS    CL28           ACCESS MATCHING KEY                       02510000
         DS    CL172          FILLER                                    02520000
W2END    EQU   *                                                        02530000
W2LENGTH EQU   W2END-W2BAL    LENGTH OF W2BAL                           02540000
```

W2BALL COPY File

```
*/*********************************************************************/ 00000100
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE W2BALL,   ***/ 00000200
*/*****  CONTAINING THE LAYOUT OF THE OPTIONAL LONG WORK AREA 2    ***/ 00000300
*/*****  FOR FUNCTIONS 1 AND 3.                                    ***/ 00000400
*/*********************************************************************/ 00000500
*/*****         LAST UPDATED  7 FEBRUARY 2006                      ***/ 00000603
*/*********************************************************************/ 00000702
W2BALL   DS   0H                                                        00000802
W2LACKEY DS    CL21          ACCESS KEY                                 00100000
W21LCPAR DS     CL1          CONTINUOUS PARITY INDICATOR                00120002
W21LLHNI DS    0CL6          LOW HOUSE NUMBER                           00130000
W21LHSEL DS     CL5          LOW HOUSE NUMBER ON BLOCK FACE             00140000
W21LSFXL DS     CL1          LOW HOUSE NUMBER SUFFIX                    00150013
W21LHHNI DS    0CL6          HIGH HOUSE NUMBER                          00160000
W21LHSEH DS     CL5          HIGH HOUSE NUMBER ON BLOCK FACE            00170000
W21LSFXH DS     CL1          HI HOUSE NUMBER SUFFIX                     00180013
W21LALX  DS    CL1           A=ALLEYS INTERSECT SEGMENT                 00190000
*                            X=CROSS STREETS MODIFIED
W21L#STL DS    CL1           NUMBER OF CROSS STREETS AT LOW END         00200000
W21LCDEL DS    CL20          UP TO FIVE PB5SC'S FOR LOW END             00210000
W21L#STH DS    CL1           NUMBER OF CROSS STREETS AT HIGH END        00220000
W21LCDEH DS    CL20          UP TO FIVE PB5SC'S FOR HIGH END            00230000
W21LCD   DS    0CL3          COMMUNITY DISTRICT                         00240000
W21LCDB  DS     CL1          COMMUNITY DISTRICT BORO                    00250000
W21LCDN  DS     CL2          COMMUNITY DISTRICT NUMBER                  00260000
W21LZIP  DS    CL5           ZIP CODE                                   00270000
W21LSLA  DS    CL1           STREET LIGHT AREA                          00280000
W21LHCD  DS    CL2           HEALTH CODE DISTRICT                       00290000
W21LSOS  DS    CL1           SIDE OF STREET INDICATOR                   00300000
W21LPAR  DS    CL1           CONTINUOUS PARITY INDICATOR                00310000
W21LCT00 DS    CL6           2000 CENSUS TRACT                          00320015
W21LCB00 DS    CL4           2000 CENSUS BLOCK                          00350015
W21LINSD DS    CL2           INSTRUCTIONAL DIVISION                     00351015
         DS    CL2           FILLER                                     00352015
W21LHA   DS    CL4           HEALTH AREA                                00360000
W21LSREC DS    CL3           SANITATION RECYCLE PICK-UP                 00370007
W21LFEAT DS    CL1           FEATURE TYPE CODE                          00380007
RES1L    DS    CL1           RESERVED FOR DCP/GSS USE                   00400000
W21LCURV DS    CL1           CURVE FLAG                                 00410014
W21LPOL  DS    0CL4          POLICE DISTRICT                            00420000
W21LPBC  DS     CL1          POLICE PATROL BORO COMMAND                 00430000
W21LPOP  DS     CL3          POLICE PRECINCT                            00440000
W21LSCH  DS    CL2           SCHOOL DISTRICT                            00450000
         DS    CL15         RESERVED FOR POLITICAL INFORMATION          00460013
W21LSTC  DS    CL1           SEGMENT TYPE CODE
W21LSAND DS    CL3           SANITATION DISTRICT                        00470002
W21LSANT DS    CL2           SANITATION DEPT SUBSECTION                 00480000
W21LFS   DS    CL2           FIRE DIVISION                              00490001
W21LFB   DS    CL2           FIRE BATTALION                             00500000
W21LFC   DS    0CL4          FIRE COMPANY                               00510000
W21LFCT  DS     CL1          FIRE COMPANY TYPE                          00520000
W21LFCN  DS     CL3          FIRE COMPANY NUMBER                        00530000
W21LSPAD DS    CL1           SPECIAL ADDRESS FLAG                       00540000
W21LMHRI DS    CL1           MARBLE HILL/RIKERS ISLAND FLAG             00550000
W21LSSCH DS    CL1           SPLIT SCHOOL DISTRICT FLAG                 00560000
W21LLGC  DS    CL2           LOGICAL GROUP CODE (PREFERRED)             00570000
W21LFACE DS    CL4           LION FACE CODE                             00580000
W21LSEQ  DS    CL5           LION SEQUENCE NUMBER                       00590000
W21LCT90 DS    CL6           1990 CENSUS TRACT                          00600000
         DS    CL4           FILLER                                     00610015
W21LCPB  DS    CL3           DYNAMIC  BLOCK                             00640000
W21LXCOR DS    CL7           X COORDINATE                               00650000
W21LYCOR DS    CL7           Y COORDINATE                               00660000
W21LSEGL DS    CL5           SEGMENT LEGNTH                             00670000
W21LSREG DS    CL5           SANITATION REGULAR PICK-UP                 00680012
```

W2BALL COPY File (continued)

```
W21LSEGT DS    CL7            SEGMENT IDENTIFIER
W21LB7SC DS    CL8            "TRUE" BOROUGH AND 7 DIGIT STREET CODE
W21LHNI  DS    CL6            UNDERLYING HOUSE NUMBER
         DS    CL79           FILLER - FUTURE USE
*                                                                       00690000
*********************************************************************** 00700000
         ORG   W21LSCH+2      PATCH FOR FUNCTION 1E FIELDS              00710000
*********************************************************************** 00720000
*                                                                       00730000
W21LEED  DS    CL3            ELECTION DISTRICT                         00740000
W21LEAD  DS    CL2            ASSEMBLY DISTRICT                         00750000
W21LESED DS    CL1            SPLIT E.D. FLAG                           00760000
W21LECON DS    CL2            CONGRESSIONAL DISTRICT                    00770000
W21LESEN DS    CL2            SENATORIAL DISTRICT                       00780000
W21LECIV DS    CL2            CIVIL COURT DISTRICT                      00790000
W21LECOU DS    CL2            CITY COUNCIL DISTRICT                     00800000
         DS    CL18                                                     00810000
W21LELGC DS    CL2            LOGICAL GROUP CODE (PREFERRED)            00820000
*                                                                       00830000
*********************************************************************** 00840000
         ORG   W2LACKEY       RESET LOCATION COUNTER FOR FUNCTION 3     00850000
*********************************************************************** 00860000
*                                                                       00870000
         DS    CL21                                                     00000902
W23LDUPF DS    CL1            DUPLICATE KEY FLAG                        00001002
W23LCURV DS    CL1            CURVE FLAG                                00001103
W23LLST  DS    CL1            LOCATION STATUS OF SEGMENT
W23LCBI  DS    CL1            COUNTY BOUNDARY INDICATOR
         DS    CL4                                                      00001203
W23LLGC1 DS    CL2            STREET 1 PREFERRED LGC                    00001302
W23LLGC2 DS    CL2            STREET 2 PREFERRED LGC                    00001402
W23LLGC3 DS    CL2            STREET 3 PREFERRED LGC                    00001502
W23L#STL DS    CL1            NUMBER OF CROSS STREETS AT LOW END        00001602
W23LCDEL DS    CL20           CROSS STREET PB5SC'S AT LOW END           00001702
W23L#STH DS    CL1            NUMBER OF CROSS STREETS AT HIGH END       00001802
W23LCDEH DS    CL20           CROSS STREET PB5SC'S AT HIGH END          00001902
W23LSLA  DS    CL1            STREET LIGHT AREA                         00002002
W23LREVF DS    CL1            REVERSAL FLAG                             00002102
W23LCDL  DS    0CL3           LEFT COMMUNITY DISTRICT                   00002202
W23LCDBL DS     CL1           LEFT COMMUNITY DISTRICT BORO              00002302
W23LCDNL DS     CL2           LEFT COMMUNITY DISTRICT NUMBER            00002402
W23LCDR  DS    0CL3           RIGHT COMMUNITY DISTRICT                  00002502
W23LCDBR DS     CL1           RIGHT COMMUNITY DISTRICT BORO             00002602
W23LCDNR DS     CL2           RIGHT COMMUNITY DISTRICT NUMBER           00002702
W23LZIPL DS    CL5            LEFT ZIP CODE                             00002802
W23LZIPR DS    CL5            RIGHT ZIP CODE                            00002902
         DS    CL18
W23LHAL  DS    CL4            LEFT HEALTH AREA                          00003602
W23LHAR  DS    CL4            RIGHT HEALTH AREA                         00003702
W23LINSL DS    CL2            LEFT INSTRUCTIONAL DIVISION               00003802
W23LINSR DS    CL2            RIGHT INSTRUCTIONAL DIVISION
W23LLO#L DS    CL7            LEFT LOW HOUSE NUMBER                     00003902
W23LHI#L DS    CL7            LEFT HIGH HOUSE NUMBER                    00004002
W23LLO#R DS    CL7            RIGHT LOW HOUSE NUMBER                    00004102
W23LHI#R DS    CL7            RIGHT HIGH HOUSE NUMBER                   00004202
W23LPAR  DS    CL1            CONTINUOUS PARITY INDICATOR               00004302
W23LFACE DS    CL4            LION FACE CODE                            00004402
W23LSEQ  DS    CL5            LION SEQUENCE NUMBER                      00004502
W23LGEN  DS    CL1            GENERATED RECORD FLAG                     00004602
W23LSEGL DS    PL3            SEGMENT LENGTH IN FEET                    00004702
W23LSLOP DS    CL3            SEGMENT SLOPE IN DEGREES                  00004802
W23LORNT DS    CL1            SEGMENT ORIENTATION                       00004902
         DS    CL4            FILLER                                    00005002
RESL1    DS    CL2            RESERVED FOR DCP/GSS USE                  00005102
W23LDGLG DS    CL1            DOG LEG FLAG
W23LFEAT DS    CL1            FEATURE TYPE CODE                         00005202
```

W2BALL COPY File (continued)

```
W23LPOLL DS    0CL4          LEFT POLICE DISTRICT                       00005302
W23LPBCL DS     CL1          LEFT POLICE PATROL BORO COMMAND            00005402
W23LPOPL DS     CL3          LEFT POLICE PRECINCT                       00005502
W23LPOLR DS    0CL4          RIGHT POLICE DISTRICT                      00005602
W23LPBCR DS     CL1          RIGHT POLICE PATROL BORO COMMAND           00005702
W23LPOPR DS     CL3          RIGHT POLICE PRECINCT                      00005802
W23LSCHL DS    CL2           LEFT SCHOOL DISTRICT                       00005902
W23LSCHR DS    CL2           RIGHT SCHOOL DISTRICT                      00006002
W23LMHRI DS    CL1           MARBLE HILL / RIKERS ISLAND                00006102
W23LSEGT DS    CL7           SEGMENT IDENTIFIER
W23LSTC  DS    CL1           SEGMENT TYPE CODE                          00006202
W23LT90L DS    CL6           1990 LEFT CENSUS TRACT                     00006302
         DS    CL4           FILLER
W23LCPBL DS    CL3           CURRENT LEFT DYNAMIC  BLOCK                00006702
W23LT90R DS    CL6           1990 RIGHT CENSUS TRACT                    00006802
         DS    CL4           FILLER
W23LCPBR DS    CL3           CURRENT RIGHT DYNAMIC  BLOCK               00007202
W23LFSL  DS    CL2           LEFT FIRE DIVISION                         00007302
W23LFBL  DS    CL2           LEFT FIRE BATTALION                        00007402
W23LFCL  DS    0CL4          LEFT FIRE COMPANY                          00007502
W23LFCTL DS     CL1          LEFT FIRE COMPANY TYPE                     00007602
W23LFCNL DS     CL3          LEFT FIRE COMPANY NUMBER                   00007702
W23LFSR  DS    CL2           RIGHT FIRE DIVISION                        00007802
W23LFBR  DS    CL2           RIGHT FIRE BATTALION                       00007902
W23LFCR  DS    0CL4          RIGHT FIRE COMPANY                         00008002
W23LFCTR DS     CL1          RIGHT FIRE COMPANY TYPE                    00008102
W23LFCNR DS     CL3          RIGHT FIRE COMPANY NUMBER                  00008202
W23LT00L DS    CL6           LEFT 2000 CENSUS TRACT
W23LB00L DS    CL4           LEFT 2000 CENSUS BLOCK
         DS    CL1           POSSIBLE CENSUS BLOCK SUFFIX
W23LT00R DS    CL6           RIGHT 2000 CENSUS TRACT
W23LB00R DS    CL4           RIGHT 2000 CENSUS BLOCK
         DS    CL1           POSSIBLE CENSUS BLOCK SUFFIX
         DS    CL36          FILLER                                     00008302
W23LEND  EQU   *                                                        00008402
W23LLEN  EQU   W23LEND-W2BALL    LENGTH OF W2BALL                       00008502
```

W2BAL1A COPY File

```
*/*******************************************************************/ 00000100
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE W2BAL1A,  ***/ 00000200
*/*****  CONTAINING THE LAYOUT OF WORK AREA 2 FOR FUNCTION         ***/ 00000300
*/*****  1A AND BL WHICH SHARE A SINGLE WORK AREA 2 LAYOUT.        ***/ 00000400
*/*******************************************************************/ 00000500
*/*****          LAST UPDATED 2 APRIL 2002                         ***/ 00000606
*/*******************************************************************/ 00000703
W2BAL1A  DS   0H                                                        00000803
         DS    CL21                                                     00000903
W21ACPAR DS    CL1            CONTINUOUS PARITY INDICATOR               00001003
W21AHSEL DS    CL6            LOW HOUSE NUMBER ON BLOCK                 00001103
W21AALT1 DS   0CL10           ALTERNATE KEY                             00001203
W21ABOR1 DS    CL1            ALTERNATE KEY - BORO                      00001303
W21ATXB1 DS    CL5            ALTERNATE KEY - TAX BLOCK                 00001403
W21ATXL1 DS    CL4            ALTERNATE KEY - TAX LOT                   00001503
         DS    CL1               FILLER                                 00001603
W21ARSCC DS    CL1            RPAD SCC                                  00001703
         DS    CL1            FILLER                                    00001803
W21AGLI  DS   0CL11           GENERAL LOT INFO                          00001903
W21ARBLC DS    CL2            RPAD BUILDING CLASSIFICATION              00002003
W21ACORC DS    CL2            CORNER CODE                               00002103
W21A#STC DS    CL2            TOTAL NUMBER STRUCTURES                   00002203
W21A#BFA DS    CL2            TOTAL NUMBER BLOCKFACES                   00002303
W21AINTF DS    CL1            INTERIOR LOT FLAG                         00002403
W21AVACF DS    CL1            VACANT LOT FLAG                           00002503
W21AIRLF DS    CL1            IRREGULARLY-SHAPED LOT FLAG               00002603
W21AMHRI DS   0CL1            MARBLE HILL/RIKERS ISLAND FLAG            00002703
W21AABFL DS    CL1            ALTERNATE BORO FLAG                       00002803
         DS    CL1            STROLLING FLAG  (W21ASTRF)                00002903
W21ASTRK DS    CL13           STROLLING KEY                             00003003
W21AOVFL DS    CL1            ADDRESS RANGE LIST OVERFLOW FLAG          00003103
W21ARFIU DS    CL1            RESERVED FOR INTERNAL USE                 00003203
W21ABIN  DS    CL7            BUILDING IDENTIFICATION NUMBER (BIN)      00003303
W21ACONF DS    CL1            CONDO LOT FLAG                            00003403
W21ARCO# DS    CL4            RPAD CONDO NUMBER                         00003503
W21ACLBL DS    CL10           CONDO LOW BBL                             00003603
         DS    CL1               FILLER                                 00003703
W21ACBBL DS    CL10           CONDO BILLING BBL                         00003803
         DS    CL1               FILLER                                 00003903
W21ACBBS DS    CL1            CONDO BILLING BBL SCC                     00004003
W21ACHBL DS    CL10           CONDO HIGH BBL                            00004103
         DS    CL1               FILLER                                 00004203
W21ASBVP DS    CL8            SANDBORN BOROUGH/VOLUME/PAGE              00004303
W21ABUSA DS    CL5            BUSINESS AREA                             00004403
W21ACOOP DS    CL4            COOP ID NUMBER                            00004503
         DS    CL4               FILLER                                 00004605
W21ANBST DS    CL4            ACTUAL TOTAL NBR OF BLDGS ON LOT          00004704
W21ATAXB DS    CL1            TAX MAP BOROUGH                           00004805
W21ATAXM DS    CL4            TAX MAP NBR - SECTION AND VOLUME          00004905
         DS    CL4            RESERVED FOR TAX MAP PAGE NUMBER          00005005
W21AXCO  DS    CL7            X COORDINATE OF ANNOTATION POINT          00005107
W21AYCO  DS    CL7            Y COORDINATE OF ANNOTATION POINT          00005207
         DS    CL18              FILLER                                 00005307
W21A#ADR DS    CL2            TOTAL ADDRESSES FOR LOT                   00005403
W21ALIST DS   0CL756          LIST OF ADDRESSES, MAXIMUM OF 21          00005503
W21ALOW# DS    CL6            LOW HOUSE NUMBER                          00005603
         DS    CL3               FILLER                                 00005703
W21AHI#  DS    CL6            HIGH HOUSE NUMBER                         00005803
         DS    CL3            FILLER                                    00005903
W21ACODE DS    CL8            STREET CODE                               00006003
W21ALBIN DS    CL7            LIST BIN                                  00006103
W21AATYP DS    CL1            ADDRESS TYPE                              00006203
         DS    CL1            FILLER                                    00006303
W21ALSOS DS    CL1            LIST SOS                                  00006403
*  STORAGE IS RESERVED FOR THE REMAINING 20 ADDRESS STRUCTURES.         00006503
```

W2BAL1A COPY File (continued)

```
*  EACH STRUCTURE IS IDENTICAL TO THE ONE DEFINED ABOVE.                    00006603
         DS    CL720         REMAINING ADDRESSES                            00006703
W21AEND  EQU   *                                                            00006803
W21ALEN  EQU   W21AEND-W2BAL1A    LENGTH OF W2BAL1A                         00007003
```

W2BAL1AL COPY File

```
*/*******************************************************************/ 00000100
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE W2BAL1AL, ***/ 00000200
*/*****  CONTAINING THE LONG LAYOUT OF WORK AREA 2 FOR FUNCTION    ***/ 00000300
*/*****  1A AND BL WHICH SHARE A SINGLE WORK AREA 2 LAYOUT.        ***/ 00000400
*/*******************************************************************/ 00000500
*/*****        Last Updated 8 March 2002                          ***/ 00000613
*/*******************************************************************/ 00000710
W2BAL1AL DS   0H                                                       00000810
         DS    CL21                                                    00000910
W21ALCPA DS    CL1           CONTINUOUS PARITY INDICATOR               00001010
W21ALHSE DS    CL6           LOW HOUSE NUMBER ON BLOCK                 00001110
W21ALALT DS   0CL10          ALTERNATE KEY                             00001210
W21ALBOR DS    CL1           ALTERNATE KEY - BORO                      00001310
W21ALTXB DS    CL5           ALTERNATE KEY - TAX BLOCK                 00001410
W21ALTXL DS    CL4           ALTERNATE KEY - TAX LOT                   00001510
         DS    CL1              FILLER                                 00001610
W21ALRSC DS    CL1           RPAD SCC                                  00001710
         DS    CL1           FILLER                                    00001810
W21ALGLI DS   0CL11          GENERAL LOT INFO                          00001910
W21ALRBL DS    CL2           RPAD BUILDING CLASSIFICATION              00002010
W21ALCOR DS    CL2           CORNER CODE                               00002110
W21AL#ST DS    CL2           TOTAL NUMBER STRUCTURES                   00002210
W21AL#BF DS    CL2           TOTAL NUMBER BLOCKFACES                   00002310
W21ALINT DS    CL1           INTERIOR LOT FLAG                         00002410
W21ALVAC DS    CL1           VACANT LOT FLAG                           00002510
W21ALIRL DS    CL1           IRREGULARLY-SHAPED LOT FLAG               00002610
W21ALMHR DS    0CL1          Marble Hill/Rikers Island Flag            00002710
W21ALABF DS    CL1           ALTERNATE BORO FLAG                       00002810
         DS    CL1           STROLLING FLAG  (W21ALSTRF)               00002910
W21ALSTR DS    CL13          STROLLING KEY                             00003010
         DS    CL1              FILLER                                 00003110
W21ALRFI DS    CL1           RESERVED FOR INTERNAL USE                 00003210
W21ALNGB DS    CL7           BUILDING IDENTIFICATION NUMBER (BIN)      00003310
W21ALCON DS    CL1           CONDO LOT FLAG                            00003410
W21ALRCO DS    CL4           RPAD CONDO NUMBER                         00003510
W21ALCLB DS    CL10          CONDO LOW BBL                             00003610
         DS    CL1              FILLER                                 00003710
W21ALCBB DS    CL10          CONDO BILLING BBL                         00003810
         DS    CL1              FILLER                                 00003910
W21ALCBS DS    CL1           CONDO BILLING BBL SCC                     00004010
W21ALCHB DS    CL10          CONDO HIGH BBL                            00004110
         DS    CL1              FILLER                                 00004210
W21ALSBV DS    CL8           SANDBORN BOROUGH/VOLUME/PAGE              00004310
W21ALBUS DS    CL5           BUSINESS AREA                             00004410
W21ALCOO DS    CL4           COOP ID NUMBER                            00004510
         DS    CL4                                                     00004613
W21ALNBS DS    CL4           Actual Nbr of Bldgs on Lot                00004711
W21ALTMB DS    CL1           TAX MAP BOROUGH                           00004814
W21ALTAX DS    CL4           Tax Map NBR - Section and Volume          00004913
         DS    CL4           RESERVED FOR TAX PAGE NUMBER              00005013
W21ALXCO DS    CL7           X COORDINATE OF ANNOTATION POINT          00005115
W21ALYCO DS    CL7           Y COORDINATE OF ANNOTATION POINT          00005215
         DS    CL16             FILLER                                 00005315
W21AL#BN DS    CL4           TOTAL Number of BINS for Lot              00005410
W21ALLST DS    2500CL7       LIST OF BINS, MAXIMUM OF 2500             00005509
W21ALEND EQU   *                                                       00005602
W21ALLEN EQU   W21ALEND-W2BAL1AL    Length of W2BAL1AL                 00006002
```

W2BAL3S COPY File

```
*/*******************************************************************/
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE W2BAL3S,  ***/
*/*****  CONTAINING THE LAYOUT OF WORK AREA 2 FOR FUNCTION 3S.      ***/
*/*******************************************************************/
W2BAL3S  DS   0H
W23SAKEY DS    CL21          ACCESS KEY
W23S#INT DS    CL3           NUMBER OF INTERSECTIONS ON STRETCH
W23SINT  DS   0CL12          INTERSECTION LAYOUT
W23SCDE1 DS    PL4           NUMERICALLY SMALLEST PB5SC
W23SCDE2 DS    PL4           NUMERICALLY 2ND SMALLEST PB5SC
W23SDIST DS    PL3           DISTANCE IN FEET FROM PREVIOUS INTERSECT.
W23SGAPF DS    CL1           GAP FLAG ("G" IF NO SEGMENT CONNECTS THIS
*                            INTERSECTION TO THE PREVIOUS ONE)
*
*     THE MAXIMUM NUMBER OF INTERSECTIONS IS 350. THE LAYOUT OF EACH
*  INTERSECTION IS IDENTICAL TO THE 12 BYTES DEFINED BY "W23SINT".
*  RATHER THAN DEFINE 349 MORE INTERSECTIONS, WE ALLOCATE THE STORAGE
*  NECESSSARY SHOULD THE MAXIMUM NUMBER OF INTERSECTIONS BY FOUND.
*     ALL INTERSECTIONS BUT THE FIRST ONE MUST BE REFERENCED BY
*  DISPLACEMENT.
*
W23SREST DS    CL4188        REMAINING INTERSECTIONS
W23SEND  EQU   *
W23SLEN  EQU   W23SEND-W2BAL3S     LENGTH OF W2BAL3S
```

PL/1 COPY Files (MSW)

W1PL1 COPY File

```
/*****************************************************************/
/***  THIS IS GEOSUPPORT SYSTEM COPY FILE W1PL1, CONTAINING THE ***/
/***  LAYOUT OF WORK AREA 1.    COPYLIB2             04/07/98  ***/
/*****************************************************************/
 DCL PW1 POINTER;
 DCL
   1 W1PL1,
     /*****************************************************/
     /*****               INPUT  FIELDS              *****/
     /*****************************************************/
     2 GEO_WA1_IN_FUNCTION_CODE,
       3 GEO_WA1_IN_FUNCTION_1         CHAR(1),
       3 GEO_WA1_IN_FUNCTION_2         CHAR(1),
     2 GEO_WA1_IN_BORO                 CHAR(1),
     2 GEO_WA1_IN_HOUSENUM             CHAR(12),/*HIGH HSE# INPUT*/
     2 GEO_WA1_IN_HOUSENUM_INTERNAL    CHAR(6), /*IF FUNCTION 5  */
     2 GEO_WA1_IN_STREET_1             CHAR(32),
     2 GEO_WA1_IN_STREET_2             CHAR(32),
     2 GEO_WA1_IN_STREET_3             CHAR(32),
     2 GEO_WA1_IN_COMPASS              CHAR(01),
     2 GEO_WA1_IN_COMPASS2             CHAR(01),
     2 GEO_WA1_IN_STREETCODE_1         FIXED DEC(6),
     2 GEO_WA1_IN_STREETCODE_2         FIXED DEC(6),
     2 GEO_WA1_IN_STREETCODE_3         FIXED DEC(6),
     2 GEO_WA1_IN_ROADBED_REQ_SWITCH   CHAR(1),
     2 GEO_WA1_IN_BORO_2               CHAR(1),
     2 GEO_WA1_IN_BORO_3               CHAR(1),
     2 GEO_WA1_IN_SNL                  CHAR(2),
     2 GEO_WA1_IN_10SC_1               CHAR(11),
     2 GEO_WA1_IN_10SC_2               CHAR(11),
     2 GEO_WA1_IN_10SC_3               CHAR(11),
     2 GEO_WA1_IN_CUI                  CHAR(5), /*NOT IMPLEMENTED*/
     2 GEO_WA1_IN_BBL,
       3 GEO_WA1_IN_BL_BORO            CHAR(1),
       3 GEO_WA1_IN_BLOCKNUM           CHAR(5),
       3 GEO_WA1_IN_LOTNUM             CHAR(4),
     2 FILLER_W1_010                   CHAR(1),
     2 GEO_WA1_IN_BIN                  CHAR(7),
     /**********************************************************/
     /******------ USAGE NOTES FOR SELECTED FIELDS ------*******/
     /**********************************************************/
     /** GEO_WA1_IN_COMPACT_NAME_FLAG: SET TO "C" TO REQUEST    **/
     /**     COMPACT NAMES OPTION.                              **/
     /** GEO_WA1_IN_LONG_WORKAREA2_FLAG: SET TO "L" TO REQUEST  **/
     /**     THE LONG WORKAREA 2. AT PRESENT, ONLY FUNCTIONS    **/
     /**     1A AND 3 HAVE THE LONG WA2 OPTION.                 **/
     /** GEO_WA1_IN_NON_IBM_MAIN_FRAME: SET TO "X" IF           **/
     /**     APPLICATION IS RUNNING ON A NON-IBM MAIN FRAME.    **/
     /** GEO_WA1_IN_1ABL_VERSION: SET TO "L" OR " " TO REQUEST  **/
     /**     THE LEGACY WORKAREA2 FORMAT FOR FUNCTION 1A OR BL. **/
     /** GEO_WA1_IN_1ABL_VERSION: SET TO "S" TO REQUEST THE     **/
     /**     STANDARD WORKAREA2 FORMAT FOR FUNCTION 1A OR BL.   **/
     /**********************************************************/
     2 GEO_WA1_IN_COMPACT_NAME_FLAG    CHAR(1),
     2 GEO_WA1_IN_LONG_WORKAREA2_FLAG  CHAR(1),
     2 GEO_WA1_IN_LOW_HOUSENUM         CHAR(12),
     2 GEO_WA1_IN_LOW_HSENUM_INTERNAL  CHAR(6),
     2 GEO_WA1_IN_NON_IBM_MAIN_FRAME   CHAR(1),
     2 GEO_WA1_IN_1ABL_VERSION         CHAR(1),
     2 GEO_WA1_IN_XSTREET_FLAG         CHAR(1),
     2 FILLER_W1_100                   CHAR(04),
     /*****************************************************/
     /*****               OUTPUT  FIELDS             *****/
     /*****************************************************/
     2 GEO_WA1_OUT_LOW_HOUSENUM        CHAR(12),
```

W1PL1 COPY File (continued)

```
    2 GEO_WA1_OUT_BORONAME              CHAR(9),                  00007300
    2 GEO_WA1_OUT_STREET_1              CHAR(32),                 00007400
    2 GEO_WA1_OUT_STREET_2              CHAR(32),                 00007500
    2 GEO_WA1_OUT_STREET_3              CHAR(32),                 00007600
    2 GEO_WA1_OUT_HOUSENUM              CHAR(12), /*HI-HND*/      00007709
    2 GEO_WA1_OUT_HOUSENUM_INTERNAL     CHAR(6),                  00007800
    2 FILLER_W1_200                     CHAR(7),                  00007900
    2 GEO_WA1_OUT_PB5SC_1               FIXED DEC(6),             00008006
    2 FILLER_W1_210                     CHAR(2),                  00008300
    2 GEO_WA1_OUT_PB5SC_2               FIXED DEC(6),             00008406
    2 FILLER_W1_220                     CHAR(2),                  00008800
    2 GEO_WA1_OUT_PB5SC_3               FIXED DEC(6),             00008906
    2 GEO_WA1_OUT_STREET_ATTR(3)        CHAR(1),                  00009312
    2 GEO_WA1_BROWSE                    CHAR(40),                 00009400
    2 GEO_WA1_OUT_10SC_1                CHAR(11),                 00009500
    2 GEO_WA1_OUT_10SC_2                CHAR(11),                 00009600
    2 GEO_WA1_OUT_10SC_3                CHAR(11),                 00009700
    2 GEO_WA1_OUT_CUI                   CHAR(5), /*NOT IMPLEMENTED*/ 00009800
    2 GEO_WA1_OUT_BBL,                                            00009900
      3 GEO_WA1_OUT_BL_BORO             CHAR(1),                  00010000
      3 GEO_WA1_OUT_BLOCKNUM            CHAR(5),                  00010100
      3 GEO_WA1_OUT_LOTNUM              CHAR(4),                  00010200
    2 FILLER_W1_240                     CHAR(1),                  00010300
    2 GEO_WA1_OUT_BIN                   CHAR(7),                  00010411
    2 GEO_WA1_OUT_SND_ATTR              CHAR(1), /*DCP/GSS USE*/  00010705
    2 GEO_WA1_OUT_REASON_CODE           CHAR(1),                  00010800
    2 FILLER_W1_400                     CHAR(2),                  00010900
    2 GEO_WA1_OUT_RETURN_CODE,                                    00011007
      3 GEO_WA1_OUT_RC_1                CHAR(1),                  00011107
      3 GEO_WA1_OUT_RC_2                CHAR(1),                  00011207
    2 GEO_WA1_OUT_ERROR_MESSAGE         CHAR(80),                 00011400
    2 GEO_WA1_OUT_NUM_SIMILAR_NAMES     FIXED DEC(3),             00011500
    2 GEO_WA1_OUT_SIMILAR_NAMES(10)     CHAR(32);                 00011600
                                                                  00011702
/***************************************************************/ 00011801
                                                                  00011902
DCL 1 GEO_WA1_OUT_PB_5SC_1                                        00012006
        BASED (ADDR(GEO_WA1_OUT_PB5SC_1)),                        00012106
      3 GEO_WA1_OUT_PACKBORO_NOSIGN_1   CHAR(1),                  00012206
      3 GEO_WA1_OUT_STREETCODE_1_KEY    FIXED DEC(5),             00012306
    1 GEO_WA1_OUT_PB_5SC_2                                        00012406
        BASED (ADDR(GEO_WA1_OUT_PB5SC_2)),                        00012506
      3 GEO_WA1_OUT_PACKBORO_NOSIGN_2   CHAR(1),                  00012606
      3 GEO_WA1_OUT_STREETCODE_2_KEY    FIXED DEC(5),             00012706
    1 GEO_WA1_OUT_PB_5SC_3                                        00012806
        BASED (ADDR(GEO_WA1_OUT_PB5SC_3)),                        00012906
      3 GEO_WA1_OUT_PACKBORO_NOSIGN_3   CHAR(1),                  00013006
      3 GEO_WA1_OUT_STREETCODE_3_KEY    FIXED DEC(5);             00013106
                                                                  00013202
/***************************************************************/ 00013301
                                                                  00013402
DCL GEO_WA1_OUT_GRC                     CHAR(02)                  00013507
                    BASED(ADDR(GEO_WA1_OUT_RETURN_CODE));         00013607
                                                                  00013902
/***************************************************************/ 00014001
                                                                  00014102
DCL 1 WORK1PL1   BASED(PW1)               CHAR(884);              00014201
                                                                  00014302
/***************************************************************/ 00014401
                                                                  00014502
PW1=ADDR(W1PL1);                                                  00015000
```

W2PL1 COPY File

```
/******************************************************************/   00000100
/***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2PL1, CONTAINING THE ***/   00000200
/***  LAYOUT OF WORK AREA 2 FOR FUNCTIONS: 1, 1E, 2, 2C, 3, 3C, ***/   00000300
/***  5.  PLEASE NOTE THAT FUNCTIONS 2 AND 2C SHARE A SINGLE    ***/   00000400
/***  WORK AREA 2 LAYOUT.                              12/30/97  ***/   00000500
/******************************************************************/   00000600
DCL PW2 POINTER;                                                        00000700
                                                                        00000801
DCL 1 W2PL1            CHAR(200) INIT(' ');                             00000901
                                                                        00001001
DCL                                                                     00001100
  1 GEO_WA2_FUNCTION1   BASED(PW2),                                     00001200
    2 GEO_WA2_FN1_ACCESS_KEY             CHAR(21),                      00001300
    2 GEO_WA2_FN1_CONT_PARITY            CHAR(1),                       00001400
    2 GEO_WA2_FN1_LOW_HOUSENUM_INT       CHAR(6),                       00001500
    2 GEO_WA2_FN1_HI_HOUSENUM_INT        CHAR(6),                       00001600
    2 GEO_WA2_FN1_ALX                    CHAR(1),                       00001731
    2 GEO_WA2_FN1_NUM_X_ST_LOW_END       CHAR(1),                       00001800
    2 GEO_WA2_FN1_LOW_PBSC(5)            FIXED DEC(7),                  00001900
    2 GEO_WA2_FN1_NUM_X_ST_HI_END        CHAR(1),                       00002000
    2 GEO_WA2_FN1_HI_PBSC(5)             FIXED DEC(7),                  00002100
    2 GEO_WA2_FN1_COMMUN_DIST,                                          00002200
      3 GEO_WA2_FN1_COMDIST_BORO         CHAR(1),                       00002300
      3 GEO_WA2_FN1_COMDIST_NUMBER       CHAR(2),                       00002400
    2 GEO_WA2_FN1_ZIP                    CHAR(5),                       00002500
    2 GEO_WA2_FN1_SLA                    CHAR(1),                       00002600
    2 GEO_WA2_FN1_HCD                    CHAR(2),                       00002700
    2 GEO_WA2_FN1_SOS                    CHAR(1),                       00002800
    2 GEO_WA2_FN1_CONT_PARITY_IND        CHAR(1),                       00002900
    2 GEO_WA2_FN1_2000_CENSUS_TRACT      CHAR(6),                       00003109
    2 GEO_WA2_FN1_2000_CENSUS_BLOCK      CHAR(4),                       00003209
    2 GEO_WA2_FN1_INSTRUCT_DIV           CHAR(2),                       00003319
    2 FILLER_W2_260                      CHAR(2),                       00003419
    2 GEO_WA2_FN1_HEALTHAREA             CHAR(4),                       00003500
    2 GEO_WA2_FN1_SANI_REC               CHAR(3),                       00003600
    2 GEO_WA2_FN1_FEATURE_TYPE           CHAR(1),                       00003721
    2 GEO_WA2_FN1_RESDCP /*RESERVED FOR*/  CHAR(1), /*DCP/GSS USE*/     00003800
    2 GEO_WA2_FN1_CURVE_FLAG             CHAR(1),                       00003908
    2 GEO_WA2_FN1_POLICE_DIST,                                          00004000
      3  GEO_WA2_FN1_POL_PAT_B_CMD       CHAR(1),                       00004100
      3  GEO_WA2_FN1_POL_PRECINCT        CHAR(3),                       00004200
    2 GEO_WA2_FN1_SCHOOLDIST             CHAR(2),                       00004300
    2 FILLER_W2_250                      CHAR(15), /*1E POL DIST*/      00004427
    2 GEO_WA2_FN1_SEGMENT_TYPE           CHAR(1),                       00004527
    2 GEO_WA2_FN1_SANI_DIST,                                            00004600
      3 GEO_WA2_FN1_SANIDIST_BORO        CHAR(1),                       00004700
      3 GEO_WA2_FN1_SANIDIST_NUMBER      CHAR(2),                       00004800
    2 GEO_WA2_FN1_SANITATION_SUBSEC      CHAR(2),                       00004900
    2 GEO_WA2_FN1_FIRESEC   /*FIRE DIV*/   CHAR(2),                     00005000
    2 GEO_WA2_FN1_FIREBAT                CHAR(2),                       00005100
    2 GEO_WA2_FN1_FIRECO,                                               00005200
      3 GEO_WA2_FN1_FIRECO_TYPE          CHAR(1),                       00005300
      3 GEO_WA2_FN1_FIRECO_NUM           CHAR(3),                       00005400
    2 GEO_WA2_FN1_SPECIAL_ADDR_FLAG      CHAR(1),                       00005500
    2 GEO_WA2_FN1_MARBLE_RIKERS_FLAG     CHAR(1),                       00005600
    2 GEO_WA2_FN1_SPLIT_SCHOOL_FLAG      CHAR(1),                       00005700
    2 GEO_WA2_FN1_PREFERRED_LGC          CHAR(2),                       00005800
    2 GEO_WA2_FN1_LIONFACECODE           CHAR(4),                       00005900
    2 GEO_WA2_FN1_LIONSEQ                CHAR(5),                       00006000
    2 GEO_WA2_FN1_1990_CENSUSTRACT       CHAR(6),                       00006100
    2 FILLER_W2_260B                     CHAR(4),                       00006209
    2 GEO_WA2_FN1_DYN_BLOCK              CHAR(3),                       00006400
    2 GEO_WA2_FN1_XCOORD                 CHAR(7),                       00006500
    2 GEO_WA2_FN1_YCOORD                 CHAR(7),                       00006600
    2 GEO_WA2_FN1_SEGMENTLENGTH          CHAR(5),                       00006700
```

W2PL1 COPY File (continued)

```
    2 GEO_WA2_FN1_SANI_REG                CHAR(5);                    00006807
                                                                      00006901
/******************************************************************/ 00007001
                                                                      00007101
DCL 1 GEO_WA2_FN1_LOW_HOUSE_NUM                                       00007200
        BASED(ADDR(GEO_WA2_FN1_LOW_HOUSENUM_INT)),                    00007300
      3 GEO_WA2_FN1_LOW_HOUSENUM          CHAR(5),                    00007400
      3 GEO_WA2_FN1_LOW_HOUSENUMSFX       CHAR(1);                    00007525
                                                                      00008001
DCL 1 GEO_WA2_FN1_HI_HOUSE_NUM                                        00008100
        BASED(ADDR(GEO_WA2_FN1_HI_HOUSENUM_INT)),                     00008200
      3 GEO_WA2_FN1_HI_HOUSENUM           CHAR(5),                    00008300
      3 GEO_WA2_FN1_HI_HOUSENUMSFX        CHAR(1);                    00008425
                                                                      00008501
DCL 1 GEO_WA2_FN1_COMDIST                 CHAR(3)                     00008600
        BASED(ADDR(GEO_WA2_FN1_COMMUN_DIST));                         00008700
                                                                      00008801
DCL 1 GEO_WA2_FN1_SANIDIST                 CHAR(3)                    00008900
        BASED(ADDR(GEO_WA2_FN1_SANI_DIST));                           00009000
                                                                      00009101
DCL 1 GEO_WA2_FN1_POLICEDIST              CHAR(4)                     00009200
        BASED(ADDR(GEO_WA2_FN1_POLICE_DIST));                         00009300
                                                                      00009401
/******************************************************************/ 00009801
                                                                      00009901
DCL                                                                   00010000
  1 GEO_WA2_FUNCTION2   BASED(PW2),                                   00010100
    2 GEO_WA2_FN2_ACCESS_KEY              CHAR(21),                   00010200
    2 GEO_WA2_FN2_DUPINTERFLAG            CHAR(1),                    00010300
    2 FILLER_W2_270                       CHAR(9),                    00010400
    2 GEO_WA2_FN2_PREFERRED_LGC1          CHAR(2),                    00010500
    2 GEO_WA2_FN2_PREFERRED_LGC2          CHAR(2),                    00010600
    2 GEO_WA2_FN2_NUM_OF_INTERSECTS       CHAR(1),                    00010700
    2 GEO_WA2_FN2_INTERSECT_PBSC(5)       FIXED DEC(7),               00010800
    2 GEO_WA2_FN2_COMPDIR                 CHAR(01),                   00010906
    2 GEO_WA2_FN2_LEVEL_CODES(5,2)        CHAR(01),                   00011020
    2 GEO_WA2_FN2_INSTRUCT_DIV            CHAR(02),                   00011117
    2 GEO_WA2_FN2_FIRESEC   /*FIRE DIV*/  CHAR(2),                    00011200
    2 GEO_WA2_FN2_FIREBAT                 CHAR(2),                    00011300
    2 GEO_WA2_FN2_FIRECO,                                             00011400
      3 GEO_WA2_FN2_FIRECO_TYPE           CHAR(1),                    00011500
      3 GEO_WA2_FN2_FIRECO_NUM            CHAR(3),                    00011600
    2 GEO_WA2_FN2_COMMUN_DIST,                                        00011700
      3 GEO_WA2_FN2_COMDIST_BORO          CHAR(1),                    00011800
      3 GEO_WA2_FN2_COMDIST_NUMBER        CHAR(2),                    00011900
    2 GEO_WA2_FN2_ZIP                     CHAR(5),                    00012000
    2 GEO_WA2_FN2_SLA                     CHAR(1),                    00012100
    2 GEO_WA2_FN2_2000_CENSUS_TRACT       CHAR(6),                    00012210
    2 FILLER_W2_290                       CHAR(3),                    00012300
    2 GEO_WA2_FN2_HEALTHAREA              CHAR(4),                    00012400
    2 FILLER_W2_300                       CHAR(9),                    00012504
    2 GEO_WA2_FN2_LIONNODENUM             CHAR(7),                    00012605
    2 GEO_WA2_FN2_XCOORD                  CHAR(7),                    00013000
    2 GEO_WA2_FN2_YCOORD                  CHAR(7),                    00013100
    2 FILLER_W2_320                       CHAR(4),                    00013300
    2 GEO_WA2_FN2_POLICE_DIST,                                        00013400
      3  GEO_WA2_FN2_POL_PAT_B_CMD        CHAR(1),                    00013500
      3  GEO_WA2_FN2_POL_PRECINCT         CHAR(3),                    00013600
    2 GEO_WA2_FN2_SCHOOLDIST              CHAR(2),                    00013700
    2 GEO_WA2_FN2_MARBLE_RIKERS_FLAG      CHAR(1),                    00013800
    2 GEO_WA2_FN2_1990_CENSUSTRACT        CHAR(6),                    00013900
    2 GEO_WA2_FN2_SANBORN1_BORO           CHAR(1),                    00014000
    2 GEO_WA2_FN2_SANBORN1_VOL_PAGE,                                  00014100
      3 GEO_WA2_FN2_SANBORN1_VOL_NUM      CHAR(3),                    00014200
      3 GEO_WA2_FN2_SANBORN1_PAGE_NUM     CHAR(4),                    00014300
    2 GEO_WA2_FN2_SANBORN2_BORO           CHAR(1),                    00014400
```

W2PL1 COPY File (continued)

```
    2 GEO_WA2_FN2_SANBORN2_VOL_PAGE,
      3 GEO_WA2_FN2_SANBORN2_VOL_NUM       CHAR(3),
      3 GEO_WA2_FN2_SANBORN2_PAGE_NUM      CHAR(4),
    2 FILLER_W2_330                        CHAR(38);

/*******************************************************************/

DCL 1 GEO_WA2_FN2_COMDIST                  CHAR(3)
        BASED(ADDR(GEO_WA2_FN2_COMMUN_DIST));

DCL 1 GEO_WA2_FN2_POLICEDIST               CHAR(4)
        BASED(ADDR(GEO_WA2_FN2_POLICE_DIST));

DCL 1 GEO_WA2_FN2_SANBORN1_BVOLPAGE        CHAR(8)
        BASED(ADDR(GEO_WA2_FN2_SANBORN1_BORO)),
    1 GEO_WA2_FN2_SANBORN2_BVOLPAGE        CHAR(8)
        BASED(ADDR(GEO_WA2_FN2_SANBORN2_BORO));

/*******************************************************************/

DCL
  1 GEO_WA2_FUNCTION3   BASED(PW2),
    2 GEO_WA2_FN3_ACCESS_KEY               CHAR(21),
    2 GEO_WA2_FN3_DUP_KEY_FLAG             CHAR(1),
    2 GEO_WA2_FN3_CURVE_FLAG               CHAR(1),
    2 GEO_WA2_FN3_LOCATION_STATUS          CHAR(1),
    2 GEO_WA2_FN3_COUNTY_BOUNDARY          CHAR(1),
    2 FILLER_W2_340                        CHAR(4),
    2 GEO_WA2_FN3_PREFERRED_LGC1           CHAR(2),
    2 GEO_WA2_FN3_PREFERRED_LGC2           CHAR(2),
    2 GEO_WA2_FN3_PREFERRED_LGC3           CHAR(2),
    2 GEO_WA2_FN3_NUM_X_ST_LOW_END         CHAR(1),
    2 GEO_WA2_FN3_LOW_PBSC(5)              FIXED DEC(7),
    2 GEO_WA2_FN3_NUM_X_ST_HI_END          CHAR(1),
    2 GEO_WA2_FN3_HI_PBSC(5)               FIXED DEC(7),
    2 GEO_WA2_FN3_SLA                      CHAR(1),
    2 GEO_WA2_FN3_REVERSALFLAG             CHAR(1),
    2 GEO_WA2_FN3_LEFT_COMMUN_DIST,
      3 GEO_WA2_FN3_LEFT_COMDIST_BORO      CHAR(1),
      3 GEO_WA2_FN3_LEFT_COMDIST_NUM       CHAR(2),
    2 GEO_WA2_FN3_RIGHT_COMMUN_DIST,
      3 GEO_WA2_FN3_RIGHT_COMDIST_BORO     CHAR(1),
      3 GEO_WA2_FN3_RIGHT_COMDIST_NUM      CHAR(2),
    2 GEO_WA2_FN3_LEFT_ZIP                 CHAR(5),
    2 GEO_WA2_FN3_RIGHT_ZIP                CHAR(5),
    2 FILLER_WA2_350A                      CHAR(18),
    2 GEO_WA2_FN3_LEFT_HEALTHAREA          CHAR(4),
    2 GEO_WA2_FN3_RIGHT_HEALTHAREA         CHAR(4),
    2 GEO_WA2_FN3_LEFT_INSTRUCT_DIV        CHAR(2),
    2 GEO_WA2_FN3_RIGHT_INSTRUCT_DIV       CHAR(2),
    2 GEO_WA2_FN3_LEFT_LOW_HOUSENUM        CHAR(7),
    2 GEO_WA2_FN3_LEFT_HI_HOUSENUM         CHAR(7),
    2 GEO_WA2_FN3_RIGHT_LOW_HOUSENUM       CHAR(7),
    2 GEO_WA2_FN3_RIGHT_HI_HOUSENUM        CHAR(7),
    2 GEO_WA2_FN3_CONT_PARITY_IND          CHAR(1),
    2 GEO_WA2_FN3_LIONFACECODE             CHAR(4),
    2 GEO_WA2_FN3_LIONSEQ                  CHAR(5),
    2 GEO_WA2_FN3_GENRECFLAG               CHAR(1),
    2 GEO_WA2_FN3_SEGMENTLENGTH            FIXED DEC(5),
    2 GEO_WA2_FN3_SEGMENTSLOPE             CHAR(3),
    2 GEO_WA2_FN3_SEGMENTORIENT            CHAR(1),
    2 FILLER_W2_355                        CHAR(4),
    2 GEO_WA2_FN3_RESDCP /*RESERVED FOR */ CHAR(2), /*DCP/GSS USE*/
    2 GEO_WA2_FN3_DOG_LEG                  CHAR(1),
    2 GEO_WA2_FN3_FEATURE_TYPE             CHAR(1),
    2 GEO_WA2_FN3_LEFT_POLICE_DIST,
```

W2PL1 COPY File (continued)

```
      3  GEO_WA2_FN3_LEFT_POL_PAT_B_CMD    CHAR(1),                     00021400
      3  GEO_WA2_FN3_LEFT_POL_PRECINCT      CHAR(3),                    00021500
    2 GEO_WA2_FN3_RIGHT_POLICE_DIST,                                    00021600
      3  GEO_WA2_FN3_RIGHT_POL_PAT_B_CMD   CHAR(1),                     00021700
      3  GEO_WA2_FN3_RIGHT_POL_PRECINCT      CHAR(3),                   00021800
    2 GEO_WA2_FN3_LEFT_SCHLDIST            CHAR(2),                     00021900
    2 GEO_WA2_FN3_RIGHT_SCHLDIST           CHAR(2),                     00022000
    2 GEO_WA2_FN3_MARBLE_RIKERS_FLAG       CHAR(1),                     00022100
    2 GEO_WA2_FN3_SEGMENT_ID               CHAR(7),                     00022211
    2 GEO_WA2_FN3_SEGMENT_TYPE             CHAR(1);                     00022329
                                                                        00022501
/*******************************************************************/ 00022601
                                                                        00022701
DCL 1 GEO_WA2_FN3_LEFT_COMDIST                  CHAR(3)                 00022800
        BASED(ADDR(GEO_WA2_FN3_LEFT_COMMUN_DIST));                      00022900
                                                                        00023001
DCL 1 GEO_WA2_FN3_RIGHT_COMDIST                 CHAR(3)                 00023100
        BASED(ADDR(GEO_WA2_FN3_RIGHT_COMMUN_DIST));                     00023200
                                                                        00023301
DCL 1 GEO_WA2_FN3_LEFT_POLICEDIST               CHAR(4)                 00023400
        BASED(ADDR(GEO_WA2_FN3_LEFT_POLICE_DIST));                      00023500
                                                                        00023601
DCL 1 GEO_WA2_FN3_RIGHT_POLICEDIST               CHAR(4)                00023700
        BASED(ADDR(GEO_WA2_FN3_RIGHT_POLICE_DIST));                     00023800
                                                                        00023901
DCL                                                                     00024000
  1 GEO_WA2_FUNCTION3C  BASED(PW2),                                     00024100
    2 GEO_WA2_FN3C_ACCESS_KEY             CHAR(21),                     00024200
    2 GEO_WA2_FN3C_CURVE_FLAG             CHAR(1),                      00024308
    2 GEO_WA2_FN3C_SEGMENT_TYPE           CHAR(1),                      00024430
    2 GEO_WA2_FN3C_LOCATION_STATUS        CHAR(1),                      00024630
    2 GEO_WA2_FN3C_COUNTY_BOUNDARY        CHAR(1),                      00024730
    2 FILLER_W2_380                       CHAR(4),                      00024830
    2 GEO_WA2_FN3C_PREFERRED_LGC1         CHAR(2),                      00024930
    2 GEO_WA2_FN3C_PREFERRED_LGC2         CHAR(2),                      00025030
    2 GEO_WA2_FN3C_PREFERRED_LGC3         CHAR(2),                      00025130
    2 GEO_WA2_FN3C_NUM_X_ST_LOW_END       CHAR(1),                      00025230
    2 GEO_WA2_FN3C_LOW_PBSC(5)            FIXED DEC(7),                 00025330
    2 GEO_WA2_FN3C_NUM_X_ST_HI_END        CHAR(1),                      00025430
    2 GEO_WA2_FN3C_HI_PBSC(5)             FIXED DEC(7),                 00025530
    2 GEO_WA2_FN3C_COMMUN_DIST,                                         00025630
      3 GEO_WA2_FN3C_COMDIST_BORO         CHAR(1),                      00025730
      3 GEO_WA2_FN3C_COMDIST_NUMBER       CHAR(2),                      00025830
    2 GEO_WA2_FN3C_ZIP                    CHAR(5),                      00025930
    2 GEO_WA2_FN3C_SLA                    CHAR(1),                      00026030
    2 FILLER_W2_390                       CHAR(7),                      00026130
    2 GEO_WA2_FN3C_2000_CENSUS_TRACT      CHAR(6),                      00026230
    2 GEO_WA2_FN3C_2000_CENSUS_BLOCK      CHAR(4),                      00026330
    2 FILLER_W2_390B_RESV_DCP             CHAR(1),                      00026430
    2 GEO_WA2_FN3C_HEALTHAREA             CHAR(4),                      00026530
    2 GEO_WA2_FN3C_REVERSALFLAG           CHAR(1),                      00026630
    2 GEO_WA2_FN3C_SOS                    CHAR(1),                      00026730
    2 GEO_WA2_FN3C_FIRESEC  /*FIRE DIV*/  CHAR(2),                      00026830
    2 GEO_WA2_FN3C_FIREBAT                CHAR(2),                      00026930
    2 GEO_WA2_FN3C_FIRECO,                                              00027030
      3 GEO_WA2_FN3C_FIRECO_TYPE          CHAR(1),                      00027130
      3 GEO_WA2_FN3C_FIRECO_NUM           CHAR(3),                      00027230
    2 GEO_WA2_FN3C_SEGMENT_ID             CHAR(7),                      00027330
    2 GEO_WA2_FN3C_LOW_HOUSENUM           CHAR(7),                      00027430
    2 GEO_WA2_FN3C_HI_HOUSENUM            CHAR(7),                      00027530
    2 GEO_WA2_FN3C_LOW_HOUSENUM2          CHAR(7),                      00027630
    2 GEO_WA2_FN3C_HI_HOUSENUM2           CHAR(7),                      00027730
/*   2 HOUSENUM2 ONLY PRESENT IF ODD & EVEN RANGES ARE ON     */        00027830
/*   2 SAME SIDE OF STREET                                    */        00027930
    2 GEO_WA2_FN3C_CONT_PARITY_IND        CHAR(1),                      00028030
    2 GEO_WA2_FN3C_LIONFACECODE           CHAR(4),                      00028130
```

W2PL1 COPY File (continued)

```
    2 GEO_WA2_FN3C_LIONSEQ              CHAR(5),                    00028230
    2 GEO_WA2_FN3C_GENRECFLAG           CHAR(1),                    00028330
    2 GEO_WA2_FN3C_SEGMENTLENGTH        FIXED DEC(5),               00028430
    2 GEO_WA2_FN3C_SEGMENTSLOPE         CHAR(3),                    00028530
    2 GEO_WA2_FN3C_SEGMENTORIENT        CHAR(1),                    00028630
    2 GEO_WA2_FN3C_INSTRUCT_DIV         CHAR(2),                    00028730
    2 GEO_WA2_FN3C_RESDCP /*RESERVED FOR*/ CHAR(1), /*DCP/GSS USE*/ 00028830
    2 GEO_WA2_FN3C_FEATURE_TYPE         CHAR(1),                    00028930
    2 GEO_WA2_FN3C_POLICE_DIST,                                     00029030
      3  GEO_WA2_FN3C_POL_PAT_B_CMD     CHAR(1),                    00029130
      3  GEO_WA2_FN3C_POL_PRECINCT      CHAR(3),                    00029230
    2 GEO_WA2_FN3C_SCHOOLDIST           CHAR(2),                    00029330
    2 GEO_WA2_FN3C_MARBLE_RIKERS_FLAG   CHAR(1),                    00029430
    2 GEO_WA2_FN3C_1990_CENSUSTRACT     CHAR(6),                    00029530
    2 FILLER_W2_410B                    CHAR(4),                    00029630
    2 GEO_WA2_FN3C_DYN_BLOCK            CHAR(3),                    00029730
    2 FILLER_W2_420                     CHAR(5);                    00029830
                                                                    00029930
/*******************************************************************/ 00030030
                                                                    00030130
DCL 1 GEO_WA2_FN3C_COMDIST              CHAR(3)                     00030230
        BASED(ADDR(GEO_WA2_FN3C_COMMUN_DIST));                      00030330
DCL 1 GEO_WA2_FN3C_POLICEDIST           CHAR(4)                     00030430
        BASED(ADDR(GEO_WA2_FN3C_POLICE_DIST));                      00030530
                                                                    00030601
/*******************************************************************/ 00030701
                                                                    00030801
DCL                                                                 00030900
  1 GEO_WA2_FUNCTION1E  BASED(PW2),                                 00031000
    2 GEO_WA2_FN1E_ACCESS_KEY           CHAR(21),                   00031100
    2 GEO_WA2_FN1E_CONT_PARITY          CHAR(1),                    00031200
    2 GEO_WA2_FN1E_LOW_HOUSENUM_INT     CHAR(6),                    00031300
    2 GEO_WA2_FN1E_HI_HOUSENUM_INT      CHAR(6),                    00031400
    2 FILLER_W2_435                     CHAR(1),                    00031500
    2 GEO_WA2_FN1E_NUM_X_ST_LOW_END     CHAR(1),                    00031600
    2 GEO_WA2_FN1E_LOW_PBSC(5)          FIXED DEC(7),               00031700
    2 GEO_WA2_FN1E_NUM_X_ST_HI_END      CHAR(1),                    00031800
    2 GEO_WA2_FN1E_HI_PBSC(5)           FIXED DEC(7),               00031900
    2 GEO_WA2_FN1E_COMMUN_DIST,                                     00032000
      3 GEO_WA2_FN1E_COMDIST_BORO       CHAR(1),                    00032100
      3 GEO_WA2_FN1E_COMDIST_NUMBER     CHAR(2),                    00032200
    2 GEO_WA2_FN1E_ZIP                  CHAR(5),                    00032300
    2 GEO_WA2_FN1E_SLA                  CHAR(1),                    00032400
    2 GEO_WA2_FN1E_HCD                  CHAR(2),                    00032500
    2 GEO_WA2_FN1E_SOS                  CHAR(1),                    00032600
    2 GEO_WA2_FN1E_CONT_PARITY_IND      CHAR(1),                    00032700
    2 GEO_WA2_FN1E_2000_CENSUS_TRACT    CHAR(6),                    00032913
    2 GEO_WA2_FN1E_2000_CENSUS_BLOCK    CHAR(4),                    00033014
    2 GEO_WA2_FN1E_INSTRUCT_DIV         CHAR(2),                    00033124
    2 FILLER_W2_440                     CHAR(2),                    00033224
    2 GEO_WA2_FN1E_HEALTHAREA           CHAR(4),                    00033500
    2 GEO_WA2_FN1E_SANI_REC             CHAR(3),                    00033600
    2 GEO_WA2_FN1E_FEATURE_TYPE         CHAR(1),                    00033721
    2 GEO_WA2_FN1E_RESDCP /*RESERVED FOR*/ CHAR(1), /*DCP/GSS USE*/ 00033800
    2 GEO_WA2_FN1E_CURVE_FLAG           CHAR(1),                    00033908
    2 GEO_WA2_FN1E_POLICE_DIST,                                     00034000
      3  GEO_WA2_FN1E_POL_PAT_B_CMD     CHAR(1),                    00034100
      3  GEO_WA2_FN1E_POL_PRECINCT      CHAR(3),                    00034200
    2 GEO_WA2_FN1E_SCHOOLDIST           CHAR(2),                    00034300
    2 GEO_WA2_FN1E_ELECTDIST            CHAR(3),                    00034400
    2 GEO_WA2_FN1E_ASSEMDIST            CHAR(2),                    00034500
    2 GEO_WA2_FN1E_SPLIT_ED_FLAG        CHAR(1),                    00034600
    2 GEO_WA2_FN1E_CONGDIST             CHAR(2),                    00034700
    2 GEO_WA2_FN1E_SENATEDIST           CHAR(2),                    00034800
    2 GEO_WA2_FN1E_COURTDIST            CHAR(2),                    00034900
    2 GEO_WA2_FN1E_COUNCILDIST          CHAR(2),                    00035000
```

W2PL1 COPY File (continued)

```
    2 FILLER_W2_470                        CHAR(2),
    2 GEO_WA2_FN1E_SANI_DIST,
      3 GEO_WA2_FN1E_SANIDIST_BORO         CHAR(1),
      3 GEO_WA2_FN1E_SANIDIST_NUMBER       CHAR(2),
    2 GEO_WA2_FN1E_SANITATION_SUBSEC       CHAR(2),
    2 GEO_WA2_FN1E_FIRESEC  /*FIRE DIV*/   CHAR(2),
    2 GEO_WA2_FN1E_FIREBAT                 CHAR(2),
    2 GEO_WA2_FN1E_FIRECO,
      3 GEO_WA2_FN1E_FIRECO_TYPE           CHAR(1),
      3 GEO_WA2_FN1E_FIRECO_NUM            CHAR(3),
    2 GEO_WA2_FN1E_SPECIAL_ADDR__FLAG      CHAR(1),
    2 GEO_WA2_FN1E_MARBLE_RIKERS_FLAG      CHAR(1),
    2 GEO_WA2_FN1E_SPLIT_SCHOOL_FLAG       CHAR(1),
    2 GEO_WA2_FN1E_PREFERRED_LGC           CHAR(2),
    2 GEO_WA2_FN1E_LIONFACECODE            CHAR(4),
    2 GEO_WA2_FN1E_LIONSEQ                 CHAR(5),
    2 GEO_WA2_FN1E_1990_CENSUSTRACT        CHAR(6),
    2 FILLER_W2_480B                       CHAR(4),
    2 GEO_WA2_FN1E_DYN_BLOCK               CHAR(3),
    2 GEO_WA2_FN1E_XCOORD                  CHAR(7),
    2 GEO_WA2_FN1E_YCOORD                  CHAR(7),
    2 GEO_WA2_FN1E_SEGMENTLENGTH           CHAR(5),
    2 GEO_WA2_FN1E_SANI_REG                CHAR(5);

/*********************************************************************/

DCL 1 GEO_WA2_FN1E_LOW_HOUSE_NUM
        BASED(ADDR(GEO_WA2_FN1E_LOW_HOUSENUM_INT)),
      3 GEO_WA2_FN1E_LOW_HOUSENUM          CHAR(5),
      3 GEO_WA2_FN1E_LOW_HOUSENUMSFX       CHAR(1);

DCL 1 GEO_WA2_FN1E_HI_HOUSE_NUM
        BASED(ADDR(GEO_WA2_FN1E_HI_HOUSENUM_INT)),
      3 GEO_WA2_FN1E_HI_HOUSENUM           CHAR(5),
      3 GEO_WA2_FN1E_HI_HOUSENUMSFX        CHAR(1);

DCL 1 GEO_WA2_FN1E_COMDIST                 CHAR(3)
        BASED(ADDR(GEO_WA2_FN1E_COMMUN_DIST));

DCL 1 GEO_WA2_FN1E_SANIDIST                  CHAR(3)
        BASED(ADDR(GEO_WA2_FN1E_SANI_DIST));

DCL 1 GEO_WA2_FN1E_POLICEDIST              CHAR(4)
        BASED(ADDR(GEO_WA2_FN1E_POLICE_DIST));

/*********************************************************************/

DCL
  1 GEO_WA2_FUNCTION5   BASED(PW2),
    2 GEO_WA2_FN5_ADDR_MATCHING_KEY        CHAR(28),
    2 FILLER_W2_210                        CHAR(172);

/*********************************************************************/

PW2=ADDR(W2PL1);
```

W2PL1L COPY File

```
/*********************************************************************/ 00000100
/***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2PL1L, CONTAINING      ***/ 00000200
/***  THE LAYOUT OF THE OPTIONAL LONG WORK AREA 2 FOR             ***/ 00000300
/***  FUNCTION 1,1E, & 3. THIS WORK AREA SHOULD BE USED ONLY WHEN ***/ 00000400
/***  FUNCTION IS CALLED WITH THE "LONG" WORK AREA2 OPTION.       ***/ 00000500
/***                                           07/23/2001    ***/ 00000600
/*********************************************************************/ 00000700
DCL PW2L POINTER;                                                       00000800
DCL 1 W2PL1L            CHAR(300) INIT(' ');                            00000900
DCL                                                                     00001000
  1 GEO_WA2_1L_FUNCTION1   BASED(PW2L),                                 00001100
    2 GEO_WA2_1L_ACCESS_KEY               CHAR(21),                     00001200
    2 GEO_WA2_1L_CONT_PARITY              CHAR(1),                      00001300
    2 GEO_WA2_1L_LOW_HOUSENUM_INT         CHAR(6),                      00001400
    2 GEO_WA2_1L_HI_HOUSENUM_INT          CHAR(6),                      00001500
    2 GEO_WA2_1L_ALX                      CHAR(1),                      00001619
    2 GEO_WA2_1L_NUM_X_ST_LOW_END         CHAR(1),                      00001700
    2 GEO_WA2_1L_LOW_PBSC(5)              FIXED DEC(7),                 00001800
    2 GEO_WA2_1L_NUM_X_ST_HI_END          CHAR(1),                      00001900
    2 GEO_WA2_1L_HI_PBSC(5)               FIXED DEC(7),                 00002000
    2 GEO_WA2_1L_COMMUN_DIST,                                           00002100
      3 GEO_WA2_1L_COMDIST_BORO           CHAR(1),                      00002200
      3 GEO_WA2_1L_COMDIST_NUMBER         CHAR(2),                      00002300
    2 GEO_WA2_1L_ZIP                      CHAR(5),                      00002400
    2 GEO_WA2_1L_SLA                      CHAR(1),                      00002500
    2 GEO_WA2_1L_HCD                      CHAR(2),                      00002600
    2 GEO_WA2_1L_SOS                      CHAR(1),                      00002700
    2 GEO_WA2_1L_CONT_PARITY_IND          CHAR(1),                      00002800
    2 GEO_WA2_1L_2000_CENSUS_TRACT        CHAR(6),                      00002900
    2 GEO_WA2_1L_2000_CENSUS_BLOCK        CHAR(4),                      00003000
    2 GEO_WA2_1L_INSTRUCT_DIV             CHAR(2),                      00003113
    2 FILLER_W2_230                       CHAR(2),                      00003213
    2 GEO_WA2_1L_HEALTHAREA               CHAR(4),                      00003500
    2 GEO_WA2_1L_SANI_REC                 CHAR(3),                      00003600
    2 GEO_WA2_1L_FEATURE_TYPE             CHAR(1),                      00003711
    2 GEO_WA2_1L_RESDCP /*RESERVED FOR*/  CHAR(1), /*DCP/GSS USE*/      00003800
    2 GEO_WA2_1L_CURVE_FLAG               CHAR(1),                      00003900
    2 GEO_WA2_1L_POLICE_DIST,                                           00004000
      3  GEO_WA2_1L_POL_PAT_B_CMD         CHAR(1),                      00004100
      3  GEO_WA2_1L_POL_PRECINCT          CHAR(3),                      00004200
    2 GEO_WA2_1L_SCHOOLDIST               CHAR(2),                      00004300
    2 FILLER_W2_250                       CHAR(15), /*1E POL DIST*/     00004417
    2 GEO_WA2_1L_SEGMENT_TYPE             CHAR(1),                      00004517
    2 GEO_WA2_1L_SANI_DIST,                                             00004600
      3 GEO_WA2_1L_SANIDIST_BORO          CHAR(1),                      00004700
      3 GEO_WA2_1L_SANIDIST_NUMBER        CHAR(2),                      00004800
    2 GEO_WA2_1L_SANITATION_SUBSEC        CHAR(2),                      00004900
    2 GEO_WA2_1L_FIRESEC   /*FIRE DIV*/   CHAR(2),                      00005000
    2 GEO_WA2_1L_FIREBAT                  CHAR(2),                      00005100
    2 GEO_WA2_1L_FIRECO,                                                00005200
      3 GEO_WA2_1L_FIRECO_TYPE            CHAR(1),                      00005300
      3 GEO_WA2_1L_FIRECO_NUM             CHAR(3),                      00005400
    2 GEO_WA2_1L_SPECIAL_ADDR_FLAG        CHAR(1),                      00005500
    2 GEO_WA2_1L_MARBLE_RIKERS_FLAG       CHAR(1),                      00005600
    2 GEO_WA2_1L_SPLIT_SCHOOL_FLAG        CHAR(1),                      00005700
    2 GEO_WA2_1L_PREFERRED_LGC            CHAR(2),                      00005800
    2 GEO_WA2_1L_LIONFACECODE             CHAR(4),                      00005900
    2 GEO_WA2_1L_LIONSEQ                  CHAR(5),                      00006000
    2 GEO_WA2_1L_1990_CENSUSTRACT         CHAR(6),                      00006100
    2 FILLER_W2_260B                       CHAR(4),                     00006200
    2 GEO_WA2_1L_DYN_BLOCK                CHAR(3),                      00006300
    2 GEO_WA2_1L_XCOORD                   CHAR(7),                      00006400
    2 GEO_WA2_1L_YCOORD                   CHAR(7),                      00006500
    2 GEO_WA2_1L_SEGMENTLENGTH            CHAR(5),                      00006600
    2 GEO_WA2_1L_SANI_REG                 CHAR(5),                      00006700
```

W2PL1L COPY File (continued)

```
    2 GEO_WA2_1L_SEGMENT_ID               CHAR(7),
    2 GEO_WA2_1L_REAL_B7SC                CHAR(08),
    2 FILLER_W2_260C                      CHAR(85);

/*******************************************************************/

DCL 1 GEO_WA2_1L_LOW_HOUSE_NUM
        BASED(ADDR(GEO_WA2_1L_LOW_HOUSENUM_INT)),
      3 GEO_WA2_1L_LOW_HOUSENUM           CHAR(5),
      3 GEO_WA2_1L_LOW_HOUSENUMSFX        CHAR(1);

DCL 1 GEO_WA2_1L_HI_HOUSE_NUM
        BASED(ADDR(GEO_WA2_1L_HI_HOUSENUM_INT)),
      3 GEO_WA2_1L_HI_HOUSENUM            CHAR(5),
      3 GEO_WA2_1L_HI_HOUSENUMSFX         CHAR(1);

DCL 1 GEO_WA2_1L_COMDIST                  CHAR(3)
        BASED(ADDR(GEO_WA2_1L_COMMUN_DIST));

DCL 1 GEO_WA2_1L_SANIDIST                  CHAR(3)
        BASED(ADDR(GEO_WA2_1L_SANI_DIST));

DCL 1 GEO_WA2_1L_POLICEDIST               CHAR(4)
        BASED(ADDR(GEO_WA2_1L_POLICE_DIST));

/*******************************************************************/

DCL
  1 GEO_WA2_1EL_FUNCTION1E  BASED(PW2L),
    2 GEO_WA2_1EL_ACCESS_KEY              CHAR(21),
    2 GEO_WA2_1EL_CONT_PARITY             CHAR(1),
    2 GEO_WA2_1EL_LOW_HOUSENUM_INT        CHAR(6),
    2 GEO_WA2_1EL_HI_HOUSENUM_INT         CHAR(6),
    2 FILLER_W2_435                        CHAR(1),
    2 GEO_WA2_1EL_NUM_X_ST_LOW_END        CHAR(1),
    2 GEO_WA2_1EL_LOW_PBSC(5)             FIXED DEC(7),
    2 GEO_WA2_1EL_NUM_X_ST_HI_END         CHAR(1),
    2 GEO_WA2_1EL_HI_PBSC(5)              FIXED DEC(7),
    2 GEO_WA2_1EL_COMMUN_DIST,
      3 GEO_WA2_1EL_COMDIST_BORO          CHAR(1),
      3 GEO_WA2_1EL_COMDIST_NUMBER        CHAR(2),
    2 GEO_WA2_1EL_ZIP                     CHAR(5),
    2 GEO_WA2_1EL_SLA                     CHAR(1),
    2 GEO_WA2_1EL_HCD                     CHAR(2),
    2 GEO_WA2_1EL_SOS                     CHAR(1),
    2 GEO_WA2_1EL_CONT_PARITY_IND         CHAR(1),
    2 GEO_WA2_1EL_2000_CENSUS_TRACT       CHAR(6),
    2 GEO_WA2_1EL_2000_CENSUS_BLOCK       CHAR(4),
    2 GEO_WA2_1EL_INSTRUCT_DIV            CHAR(2),
    2 FILLER_W2_240                       CHAR(2),
    2 GEO_WA2_1EL_HEALTHAREA              CHAR(4),
    2 GEO_WA2_1EL_SANI_REC                CHAR(3),
    2 GEO_WA2_1EL_FEATURE_TYPE            CHAR(1),
    2 GEO_WA2_1EL_RESDCP /*RESERVED FOR*/ CHAR(1), /*DCP/GSS USE*/
    2 GEO_WA2_1EL_CURVE_FLAG              CHAR(1),
    2 GEO_WA2_1EL_POLICE_DIST,
      3  GEO_WA2_1EL_POL_PAT_B_CMD         CHAR(1),
      3  GEO_WA2_1EL_POL_PRECINCT         CHAR(3),
    2 GEO_WA2_1EL_SCHOOLDIST              CHAR(2),
    2 GEO_WA2_1EL_ELECTDIST               CHAR(3),
    2 GEO_WA2_1EL_ASSEMDIST               CHAR(2),
    2 GEO_WA2_1EL_SPLIT_ED_FLAG           CHAR(1),
    2 GEO_WA2_1EL_CONGDIST                CHAR(2),
    2 GEO_WA2_1EL_SENATEDIST              CHAR(2),
    2 GEO_WA2_1EL_COURTDIST               CHAR(2),
    2 GEO_WA2_1EL_COUNCILDIST             CHAR(2),
```

W2PL1L COPY File (continued)

```
    2 FILLER_W2_470                      CHAR(2),                     00013511
    2 GEO_WA2_1EL_SANI_DIST,                                          00013600
      3 GEO_WA2_1EL_SANIDIST_BORO        CHAR(1),                     00013700
      3 GEO_WA2_1EL_SANIDIST_NUMBER      CHAR(2),                     00013800
    2 GEO_WA2_1EL_SANITATION_SUBSEC      CHAR(2),                     00013900
    2 GEO_WA2_1EL_FIRESEC  /*FIRE DIV*/  CHAR(2),                     00014000
    2 GEO_WA2_1EL_FIREBAT                CHAR(2),                     00014100
    2 GEO_WA2_1EL_FIRECO,                                             00014200
      3 GEO_WA2_1EL_FIRECO_TYPE          CHAR(1),                     00014300
      3 GEO_WA2_1EL_FIRECO_NUM           CHAR(3),                     00014400
    2 GEO_WA2_1EL_SPECIAL_ADDR__FLAG     CHAR(1),                     00014500
    2 GEO_WA2_1EL_MARBLE_RIKERS_FLAG     CHAR(1),                     00014600
    2 GEO_WA2_1EL_SPLIT_SCHOOL_FLAG      CHAR(1),                     00014700
    2 GEO_WA2_1EL_PREFERRED_LGC          CHAR(2),                     00014800
    2 GEO_WA2_1EL_LIONFACECODE           CHAR(4),                     00014900
    2 GEO_WA2_1EL_LIONSEQ                CHAR(5),                     00015000
    2 GEO_WA2_1EL_1990_CENSUSTRACT       CHAR(6),                     00015100
    2 FILLER_W2_480B                      CHAR(4),                    00015200
    2 GEO_WA2_1EL_DYN_BLOCK              CHAR(3),                     00015300
    2 GEO_WA2_1EL_XCOORD                 CHAR(7),                     00015400
    2 GEO_WA2_1EL_YCOORD                 CHAR(7),                     00015500
    2 GEO_WA2_1EL_SEGMENTLENGTH          CHAR(5),                     00015600
    2 GEO_WA2_1EL_SANI_REG               CHAR(5),                     00015700
    2 GEO_WA2_1EL_SEGMENT_ID             CHAR(7),                     00015800
    2 GEO_WA2_1EL_TRUE_B7SC              CHAR(8),                     00015909
    2 FILLER_W2_480                      CHAR(85);                    00016009
                                                                      00016100
/*******************************************************************/ 00016200
                                                                      00016300
DCL 1 GEO_WA2_1EL_LOW_HOUSE_NUM                                       00016400
        BASED(ADDR(GEO_WA2_1EL_LOW_HOUSENUM_INT)),                    00016500
      3 GEO_WA2_1EL_LOW_HOUSENUM         CHAR(5),                     00016600
      3 GEO_WA2_1EL_LOW_HOUSENUMSFX      CHAR(1);                     00016715
                                                                      00016800
DCL 1 GEO_WA2_1EL_HI_HOUSE_NUM                                        00016900
        BASED(ADDR(GEO_WA2_1EL_HI_HOUSENUM_INT)),                     00017000
      3 GEO_WA2_1EL_HI_HOUSENUM          CHAR(5),                     00017100
      3 GEO_WA2_1EL_HI_HOUSENUMSFX       CHAR(1);                     00017215
                                                                      00017300
DCL 1 GEO_WA2_1EL_COMDIST                CHAR(3)                      00017400
        BASED(ADDR(GEO_WA2_1EL_COMMUN_DIST));                         00017500
                                                                      00017600
DCL 1 GEO_WA2_1EL_SANIDIST                 CHAR(3)                    00017700
        BASED(ADDR(GEO_WA2_1EL_SANI_DIST));                           00017800
                                                                      00017900
DCL 1 GEO_WA2_1EL_POLICEDIST             CHAR(4)                      00018000
        BASED(ADDR(GEO_WA2_1EL_POLICE_DIST));                         00018100
                                                                      00018200
/*******************************************************************/ 00018300
                                                                      00018400
DCL                                                                   00018500
  1 GEO_WA2_FUNCTION3L  BASED(PW2L),                                  00018600
    2 GEO_WA2_3L_ACCESS_KEY              CHAR(21),                    00018700
    2 GEO_WA2_3L_DUP_KEY_FLAG            CHAR(1),                     00018812
    2 GEO_WA2_3L_CURVE_FLAG              CHAR(1),                     00018900
    2 GEO_WA2_3L_LOCATION_STATUS         CHAR(1),                     00019012
    2 GEO_WA2_3L_COUNTY_BOUNDARY         CHAR(1),                     00019112
    2 FILLER_W340                        CHAR(4),                     00019212
    2 GEO_WA2_3L_PREFERRED_LGC1          CHAR(2),                     00019312
    2 GEO_WA2_3L_PREFERRED_LGC2          CHAR(2),                     00019412
    2 GEO_WA2_3L_PREFERRED_LGC3          CHAR(2),                     00019512
    2 GEO_WA2_3L_NUM_X_ST_LOW_END        CHAR(1),                     00019612
    2 GEO_WA2_3L_LOW_PBSC(5)             FIXED DEC(7),                00019712
    2 GEO_WA2_3L_NUM_X_ST_HI_END         CHAR(1),                     00019812
    2 GEO_WA2_3L_HI_PBSC(5)              FIXED DEC(7),                00019912
    2 GEO_WA2_3L_SLA                     CHAR(1),                     00020012
```

W2PL1L COPY File (continued)

```
    2 GEO_WA2_3L_REVERSALFLAG          CHAR(1),                   00020112
    2 GEO_WA2_3L_LEFT_COMMUN_DIST,                                00020212
      3 GEO_WA2_3L_LEFT_COMDIST_BORO   CHAR(1),                   00020312
      3 GEO_WA2_3L_LEFT_COMDIST_NUMBER CHAR(2),                   00020412
    2 GEO_WA2_3L_RIGHT_COMMUN_DIST,                               00020512
      3 GEO_WA2_3L_RIGHT_COMDIST_BORO  CHAR(1),                   00020612
      3 GEO_WA2_3L_RIGHT_COMDIST_NUMBER CHAR(2),                  00020712
    2 GEO_WA2_3L_LEFT_ZIP              CHAR(5),                   00020812
    2 GEO_WA2_3L_RIGHT_ZIP             CHAR(5),                   00020912
    2 FILLER_W340B                     CHAR(18),                  00021012
    2 GEO_WA2_3L_LEFT_HEALTHAREA       CHAR(4),                   00021112
    2 GEO_WA2_3L_RIGHT_HEALTHAREA      CHAR(4),                   00021212
    2 GEO_WA2_3L_LEFT_INSTRUCT_DIV     CHAR(2),                   00021314
    2 GEO_WA2_3L_RIGHT_INSTRUCT_DIV    CHAR(2),                   00021414
    2 GEO_WA2_3L_LEFT_LOW_HOUSENUM     CHAR(7),                   00021612
    2 GEO_WA2_3L_LEFT_HI_HOUSENUM      CHAR(7),                   00021712
    2 GEO_WA2_3L_RIGHT_LOW_HOUSENUM    CHAR(7),                   00021812
    2 GEO_WA2_3L_RIGHT_HI_HOUSENUM     CHAR(7),                   00021912
    2 GEO_WA2_3L_CONT_PARITY_IND       CHAR(1),                   00022012
    2 GEO_WA2_3L_LIONFACECODE          CHAR(4),                   00022112
    2 GEO_WA2_3L_LIONSEQ               CHAR(5),                   00022212
    2 GEO_WA2_3L_GENRECFLAG            CHAR(1),                   00022312
    2 GEO_WA2_3L_SEGMENTLENGTH         FIXED DEC(5),              00022412
    2 GEO_WA2_3L_SEGMENTSLOPE          CHAR(3),                   00022512
    2 GEO_WA2_3L_SEGMENTORIENT         CHAR(1),                   00022612
    2 FILLER_W355                      CHAR(4),                   00022712
    2 GEO_WA2_3L_RESDCP                CHAR(2),                   00022812
    2 GEO_WA2_3L_DOG_LEG               CHAR(1),                   00022912
    2 GEO_WA2_3L_FEATURE_TYPE          CHAR(1),                   00023012
    2 GEO_WA2_3L_LEFT_POLICE_DIST,                                00023112
      3  GEO_WA2_3L_LEFT_POL_PAT_B_CMD CHAR(1),                   00023212
      3  GEO_WA2_3L_LEFT_POL_PRECINCT  CHAR(3),                   00023312
    2 GEO_WA2_3L_RIGHT_POLICE_DIST,                               00023412
      3  GEO_WA2_3L_RIGHT_POL_PAT_B_CMD CHAR(1),                  00023512
      3  GEO_WA2_3L_RIGHT_POL_PRECINCT CHAR(3),                   00023612
    2 GEO_WA2_3L_LEFT_SCHLDIST         CHAR(2),                   00023712
    2 GEO_WA2_3L_RIGHT_SCHLDIST        CHAR(2),                   00023812
    2 GEO_WA2_3L_MARBLE_RIKERS_FLAG    CHAR(1),                   00023912
    2 GEO_WA2_3L_SEGMENT_ID            CHAR(7),                   00024012
    2 GEO_WA2_3L_SEGMENT_TYPE          CHAR(1),                   00024118
/***************************************************************/ 00024312
/**  THE PORTION OF THIS WORK AREA ABOVE THIS POINT IS       **/ 00024412
/**  IDENTICAL TO THE STANDARD WORK AREA 2 FOR FUNCTION 3.   **/ 00024512
/**  THE PORTION BELOW THIS POINT IS PRESENT ONLY FOR THE    **/ 00024612
/**  LONG WORK AREA 2 OPTION.                                **/ 00024712
/***************************************************************/ 00024812
    2 GEO_WA2_3L_L_1990_CENSUSTRACT    CHAR(6),                   00024912
    2 FILLER_W370B                     CHAR(4),                   00025012
    2 GEO_WA2_L_3L_DYN_BLOCK           CHAR(3),                   00025112
    2 GEO_WA2_3L_R_1990_CENSUSTRACT    CHAR(6),                   00025212
    2 FILLER_W370C                     CHAR(4),                   00025312
    2 GEO_WA2_R_3L_DYN_BLOCK           CHAR(3),                   00025412
    2 GEO_WA2_3L_LEFT_FIRESEC          CHAR(2),/*FIRE DIV*/       00025512
    2 GEO_WA2_3L_LEFT_FIREBAT          CHAR(2),                   00025612
    2 GEO_WA2_3L_LEFT_FIRECO,                                     00025712
      3 GEO_WA2_3L_LEFT_FIRECO_TYPE    CHAR(1),                   00025812
      3 GEO_WA2_3L_LEFT_FIRECO_NUM     CHAR(3),                   00025912
    2 GEO_WA2_3L_RIGHT_FIRESEC         CHAR(2),/*FIRE DIV*/       00026012
    2 GEO_WA2_3L_RIGHT_FIREBAT         CHAR(2),                   00026112
    2 GEO_WA2_3L_RIGHT_FIRECO,                                    00026212
      3 GEO_WA2_3L_RIGHT_FIRECO_TYPE   CHAR(1),                   00026312
      3 GEO_WA2_3L_RIGHT_FIRECO_NUM    CHAR(3),                   00026412
    2 GEO_WA2_3L_L_2000_CENSUS_TRACT   CHAR(6),                   00026512
    2 GEO_WA2_3L_L_2000_CENSUS_BLOCK   CHAR(4),                   00026612
    2 FILLER_W380B_RESV                CHAR(1),                   00026712
    2 GEO_WA2_3L_R_2000_CENSUS_TRACT   CHAR(6),                   00026812
```

W2PL1L COPY File (continued)

```
    2 GEO_WA2_3L_R_2000_CENSUS_BLOCK      CHAR(4),                     00026912
    2 FILLER_W380C_RESV                   CHAR(1),                     00027012
    2 FILLER_W380                         CHAR(36);                    00027112
DCL 1 GEO_WA2_3L_LEFT_COMDIST             CHAR(3)                      00027212
        BASED(ADDR(GEO_WA2_3L_LEFT_COMMUN_DIST));                      00027312
DCL 1 GEO_WA2_3L_RIGHT_COMDIST            CHAR(3)                      00027412
        BASED(ADDR(GEO_WA2_3L_RIGHT_COMMUN_DIST));                     00027512
DCL 1 GEO_WA2_3L_LEFT_POLICEDIST          CHAR(4)                      00027612
        BASED(ADDR(GEO_WA2_3L_LEFT_POLICE_DIST));                      00027712
DCL 1 GEO_WA2_3L_RIGHT_POLICEDIST         CHAR(4)                      00027812
        BASED(ADDR(GEO_WA2_3L_RIGHT_POLICE_DIST));                     00027912
PW2L=ADDR(W2PL1L);                                                     00028012
                                                                       00028112
/*********************************************************************/ 00029012
```

W2PL11A COPY File

```
/*******************************************************************/
/***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2PL11A, CONTAINING THE ***/
/***  LAYOUT OF WORK AREA 2 FOR FUNCTIONS 1A AND BL WHICH SHARE   ***/
/***  A SINGLE WORK AREA 2 LAYOUT.                      11/28/00  ***/
/*******************************************************************/

 /*  STANDARD FORMAT  */

 DCL
   1 W2PL11A,
     2 GEO_WA2_1A_ACCESS_KEY                 CHAR(21),
     2 GEO_WA2_1A_CONT_PARITY                CHAR(1),
     2 GEO_WA2_1A_LOW_HOUSENUM               CHAR(6),
     2 GEO_WA2_1A_ALTKEY_1,
       3 GEO_WA2_1A_ALTKEY_1_BORO            CHAR(1),
       3 GEO_WA2_1A_ALTKEY_1_TAXBLOCK        CHAR(5),
       3 GEO_WA2_1A_ALTKEY_1_TAXLOT          CHAR(4),
     2 GEO_WA2_1A_FILLER_230                 CHAR(1),
     2 GEO_WA2_1A_SCC                        CHAR(1),
     2 GEO_WA2_1A_FILLER_240                 CHAR(1),
     2 GEO_WA2_1A_GENERAL_LOT_INFO,
       3 GEO_WA2_1A_RPAD_BLDG_CLASS          CHAR(2),
       3 GEO_WA2_1A_CORNER_CODE              CHAR(2),
       3 GEO_WA2_1A_NUM_OF_STRUCTURES        CHAR(2),
       3 GEO_WA2_1A_NUM_OF_BLOCKFACES        CHAR(2),
       3 GEO_WA2_1A_INTERIOR_FLAG            CHAR(1),
       3 GEO_WA2_1A_VACANT_FLAG              CHAR(1),
       3 GEO_WA2_1A_IRREG_LOT_FLAG           CHAR(1),
     2 GEO_WA2_1A_ALT_BORO_FLAG              CHAR(1),
     2 GEO_WA2_1A_FILLER_245                 CHAR(1),
     2 GEO_WA2_1A_STROLL_KEY                 CHAR(13),
     2 GEO_WA2_1A_OVERFLOW_FLAG              CHAR(1),
     2 GEO_WA2_1A_FILLER_251                 CHAR(1), /*USED FOR DCP*/
     2 GEO_WA2_1A_BIN                        CHAR(7),
     2 GEO_WA2_1A_CONDO_FLAG                 CHAR(1),
     2 GEO_WA2_1A_RPAD_CONDO_ID_NUM          CHAR(4),
     2 GEO_WA2_1A_CONDO_LOW_BBL              CHAR(10),
     2 GEO_WA2_1A_FILLER_260                 CHAR(1),
     2 GEO_WA2_1A_CONDO_BILL_BBL             CHAR(10),
     2 GEO_WA2_1A_FILLER_270                 CHAR(1),
     2 GEO_WA2_1A_CONDO_BILL_BBL_SCC         CHAR(1),
     2 GEO_WA2_1A_CONDO_HIGH_BBL             CHAR(10),
     2 GEO_WA2_1A_FILLER_275                 CHAR(1),
     2 GEO_WA2_1A_SANBORN_BORO               CHAR(1),
     2 GEO_WA2_1A_SANBORN_VOL_PAGE,
       3 GEO_WA2_1A_SANBORN_VOL_NUM          CHAR(3),
       3 GEO_WA2_1A_SANBORN_PAGE_NUM         CHAR(4),
     2 GEO_WA2_1A_COMMERC_DIST               CHAR(5),
     2 GEO_WA2_1A_COOP_NUM                   CHAR(4),
     2 GEO_WA2_1A_FILLER_276                 CHAR(4),
     2 GEO_WA2_1A_ACTUAL_NUM_OF_STRUCT       CHAR(4),
     2 GEO_WA2_1A_DOF_MAP_BORO               CHAR(1),
     2 GEO_WA2_1A_DOF_MAP_SECVOL             CHAR(4),
     2 GEO_WA2_1A_DOF_MAP_PAGE               CHAR(4),
     2 GEO_WA2_1A_X_COORD                    CHAR(7),
     2 GEO_WA2_1A_Y_COORD                    CHAR(7),
     2 GEO_WA2_1A_FILLER_280                 CHAR(18),
     2 GEO_WA2_1A_NUM_OF_ADDR_FOR_LOT        CHAR(2),
     2 GEO_WA2_1A_LIST_OF_ADDRESSES(21),
       3 GEO_WA2_1A_LIST_LOW_HOUSENUM        CHAR(6),
       3 GEO_WA2_1A_FILLER_290               CHAR(3),
       3 GEO_WA2_1A_LIST_HI_HOUSENUM         CHAR(6),
       3 GEO_WA2_1A_FILLER_300               CHAR(3),
       3 GEO_WA2_1A_LIST_STREETCODE          CHAR(8),
       3 GEO_WA2_1A_LIST_BIN                 CHAR(7),
```

W2PL11A COPY File (continued)

```
       3 GEO_WA2_1A_ADDR_TYPE                 CHAR(1),                    00006608
       3 GEO_WA2_1A_FILLER_310                CHAR(1),                    00006708
       3 GEO_WA2_1A_LIST_SOS                  CHAR(1);                    00006808
 DCL  GEO_WA2_1A_SANBORN_BVOLPAGE             CHAR(8)                     00006908
       BASED(ADDR(GEO_WA2_1A_SANBORN_BORO));                              00007008
```

W2PL11AL COPY File

```
/*********************************************************************/
/***  THIS IS GEOSUPPORT SYSTEM COPY FILE W2PL11AL, CONTAINING THE***/
/***  LAYOUT OF LONG WORK AREA 2 FOR FUNCTIONS 1A AND BL WHICH    ***/
/***  SHARE A SINGLE WORK AREA 2 LAYOUT.               11/28/00   ***/
/*********************************************************************/

 /*  1A/BL LONG WORK AREA 2 */

 DCL
   1 W2PL11AL,
     2 GEO_WA2_1AL_ACCESS_KEY                CHAR(21),
     2 GEO_WA2_1AL_CONT_PARITY               CHAR(1),
     2 GEO_WA2_1AL_LOW_HOUSENUM              CHAR(6),
     2 GEO_WA2_1AL_ALTKEY_1,
       3 GEO_WA2_1AL_ALTKEY_1_BORO           CHAR(1),
       3 GEO_WA2_1AL_ALTKEY_1_TAXBLOCK       CHAR(5),
       3 GEO_WA2_1AL_ALTKEY_1_TAXLOT         CHAR(4),
     2 GEO_WA2_1AL_FILLER_230                CHAR(1),
     2 GEO_WA2_1AL_SCC                       CHAR(1),
     2 GEO_WA2_1AL_FILLER_240                CHAR(1),
     2 GEO_WA2_1AL_GENERAL_LOT_INFO,
       3 GEO_WA2_1AL_RPAD_BLDG_CLASS         CHAR(2),
       3 GEO_WA2_1AL_CORNER_CODE             CHAR(2),
       3 GEO_WA2_1AL_NUM_OF_STRUCTURES       CHAR(2),
       3 GEO_WA2_1AL_NUM_OF_BLOCKFACES       CHAR(2),
       3 GEO_WA2_1AL_INTERIOR_FLAG           CHAR(1),
       3 GEO_WA2_1AL_VACANT_FLAG             CHAR(1),
       3 GEO_WA2_1AL_IRREG_LOT_FLAG          CHAR(1),
     2 GEO_WA2_1AL_ALT_BORO_FLAG             CHAR(1),
     2 GEO_WA2_1AL_FILLER_245                CHAR(1),
     2 GEO_WA2_1AL_STROLL_KEY                CHAR(13),
     2 GEO_WA2_1AL_FILLER_250                CHAR(1),
     2 GEO_WA2_1AL_FILLER_251                CHAR(1), /*USED FOR DCP*/
     2 GEO_WA2_1AL_BIN                       CHAR(7),
     2 GEO_WA2_1AL_CONDO_FLAG                CHAR(1),
     2 GEO_WA2_1AL_RPAD_CONDO_ID_NUM         CHAR(4),
     2 GEO_WA2_1AL_CONDO_LOW_BBL             CHAR(10),
     2 GEO_WA2_1AL_FILLER_260                CHAR(1),
     2 GEO_WA2_1AL_CONDO_BILL_BBL            CHAR(10),
     2 GEO_WA2_1AL_FILLER_270                CHAR(1),
     2 GEO_WA2_1AL_CONDO_BILL_BBL_SCC        CHAR(1),
     2 GEO_WA2_1AL_CONDO_HIGH_BBL            CHAR(10),
     2 GEO_WA2_1AL_FILLER_275                CHAR(1),
     2 GEO_WA2_1AL_SANBORN_BORO              CHAR(1),
     2 GEO_WA2_1AL_SANBORN_VOL_PAGE,
       3 GEO_WA2_1AL_SANBORN_VOL_NUM         CHAR(3),
       3 GEO_WA2_1AL_SANBORN_PAGE_NUM        CHAR(4),
     2 GEO_WA2_1AL_COMMERC_DIST              CHAR(5),
     2 GEO_WA2_1AL_COOP_NUM                  CHAR(4),
     2 GEO_WA2_1AL_FILLER_276                CHAR(4),
     2 GEO_WA2_1AL_ACTUAL_NUM_STRUCTS        CHAR(4),
     2 GEO_WA2_1AL_DOF_MAP_BORO              CHAR(1),
     2 GEO_WA2_1AL_DOF_MAP_SECVOL            CHAR(4),
     2 GEO_WA2_1AL_DOF_MAP_PAGE              CHAR(4),
     2 GEO_WA2_1AL_X_COORD                   CHAR(7),
     2 GEO_WA2_1AL_Y_COORD                   CHAR(7),
     2 GEO_WA2_1AL_FILLER_280                CHAR(16),
     2 GEO_WA2_1AL_NUM_OF_BINS_FOR_LOT       CHAR(4),
     2 GEO_WA2_1AL_BINS(2500)                CHAR(7);

DCL  GEO_WA2_1AL_SANBORN_BVOLPAGE            CHAR(8)
       BASED(ADDR(GEO_WA2_1AL_SANBORN_BORO));
```

W2PL13S COPY File

```
/*****************************************************************/
/** THIS IS GEOSUPPORT SYSTEM COPY FILE W2PL13S, CONTAINING THE   **/
/** LAYOUT OF WORK AREA 2 FOR FUNCTION 3S.               9/22/93  **/
/*****************************************************************/
 DCL
   1 W2PL13S,
     2 GEO_WA2_3S_ACCESS_KEY                CHAR(21),
     2 GEO_WA2_3S_NUM_OF_INTERSECTS         CHAR(3),
     2 GEO_WA2_3S_LIST_OF_INTERSECTS(350),
       3 GEO_WA2_3S_SMALLEST_PBSC           FIXED DEC(7),
       3 GEO_WA2_3S_2ND_SMALLEST_PBSC       FIXED DEC(7),
       3 GEO_WA2_3S_DISTANCE                FIXED DEC(5),
       3 GEO_WA2_3S_GAP_FLAG                CHAR(1);
```

C COPY File (MSW)

WAC COPY File

```
/*****************************************************************/
/*              Modified -  7 FEBRUARY 2006                      */
/*****************************************************************/
typedef struct {
         struct {
                char func_code[2];      /* Function Code             */
                char boro_1;            /* Borough Code of First St */
                char hse_nbr_disp[12];  /* House nbr in Disp form    */
                char hse_nbr_hni[6];    /* House nbr in HNI form     */
                char street_name_1[32]; /* First Street Name         */
                char street_name_2[32]; /* Second Street Name        */
                char street_name_3[32]; /* Third Street Name         */
                char comp_direction;    /* Compass Direction         */
                char comp_direction2;   /* Compass Direction-Fn 3S   */
                char PB5SC_1[4];        /* Packd Boro 5 digt St Code*/
                char PB5SC_2[4];        /* Packd Boro 5 digt St Code*/
                char PB5SC_3[4];        /* Packd Boro 5 digt St Code*/
                char roadbedreq;        /* Roadbed Request Switch    */
                char boro_2;            /* Boro Code of Second Strt */
                char boro_3;            /* Boro Code of Third Street*/
                char snl[2];            /* Street Name Norm Length   */
                char B10SC_1[11];       /* 1st Boro & 10 Digt St Cod*/
                char B10SC_2[11];       /* 2nd Boro & 10 Digt St Cod*/
                char B10SC_3[11];       /* 3rd Boro & 10 Digt St Cod*/
                char filler03[5];
                char BBL[10];           /* Boro(len=1), Block(len=5)*/
                                        /* and Lot (len=4)           */
                char filler04;
                char bld_id[7];         /* Bld Id Number (BIN)       */
                char compact_flag;      /* Compact Street Names flag*/
                char long_WA_flag;      /* Long Work Area 2 Flag     */
                char lo_range_hnd[12];  /* Low HND of Range          */
                char lo_range_hni[6];   /* Low HNI of Range          */
                char not_IBM_flag;      /* Non-IBM Mainframe Flag    */
                char BL1A;              /* 1A/BL Version Switch      */
                char xstreet_flag;      /* Cross Street Names Flag   */
                char filler06[4 ];
               } input;
         struct {
                char lo_hse_nbr_disp[12]; /* Low HND of Range        */
                char boro_name[9];      /* Boro Name of First Street*/
                char street_name_1[32]; /* 1st St Name - Normalized */
                char street_name_2[32]; /* 2nd St Name - Normalized */
                char street_name_3[32]; /* 3rd St Name - Normalized */
                char hse_nbr_disp[12];  /* House nbr in Normalized   */
                                        /* Display form              */
                char hse_nbr_hni[6];    /* House number in HNI form */
                char filler01[7];
                char PB5SC_1[4];        /* Packd Boro 5 digt St Code*/
                char filler02[2];
                char PB5SC_2[4];        /* Packd Boro 5 digt St Code*/
                char filler03[2];
                char PB5SC_3[4];        /* Packd Boro 5 digt St Code*/
                char attrbytes[3];      /* Attribute Bytes - int use*/
                char br_pb5sc[10][4];   /* Up to 10 PB5SCs-Browse fn*/
                char B10SC_1[11];       /* 1st Boro & 10 Digt St Cod*/
                char B10SC_2[11];       /* 2nd Boro & 10 Digt St Cod*/
                char B10SC_3[11];       /* 3rd Boro & 10 Digt St Cod*/
                char condo_nbr[5];      /* Condo Number              */
                char BBL[10];           /* Boro(len=1), Block(len=5)*/
                                        /* and Lot (len=4)-Normalizd*/
                char filler06[1];
                char bld_id[7];         /* Building Id Number        */
                char intuse1;           /* Internal Use Only         */
                char reject_reason_code;/* Reject Reason Code        */
```

WAC COPY File (continued)

```
                 char filler07[2];
                 char ret_code[2];        /* GeoSupport Return Code   */
                 char msg[80];            /* GeoSupport Message       */
                 char nbr_sim_names[2];   /* Nbr of Similar St Names  */
                 char sim_names[10][32];  /* Up to 10 Similar St Names*/
                } output;
               } C_WA1;

typedef struct { char filler01[21];
                 char cont_parity_ind;    /* Continuous Parity Ind.   */
                 char lo_hse_nbr[6];      /* Low House nbr in HNI form*/
                 char hi_hse_nbr[6];      /* Hi House Nbr in HNI form */
                 char alx;                /* A=Alley intersects segmnt*/
                                          /* X=Cross Streets modified */
                 char lo_nbr_x_sts;       /* Nbr of cross streets at  */
                                          /* low house nbr end of st  */
                 char l_x_sts[5][4];      /* PB5SCs of lo end cross st*/
                 char hi_nbr_x_sts;       /* Nbr of cross streets at  */
                                          /* low house nbr end of st  */
                 char h_x_sts[5][4];      /* PB5SCs of lo end cross st*/
                 char com_dist[3];        /* Community District       */
                                          /* Position 0 contains the  */
                                          /* Legacy Boro Code & Pos 1 */
                                          /* & 2, the district nbr    */
                 char zip_code[5];        /* Zip code for st seg      */
                 char DOT_slca;           /* DOT St Lght Contractr Are*/
                 char health_cent[2];     /* Health Center District   */
                 char sos_ind;            /* Side of Street Indicator */
                 char cont_par;           /* Continuous Parity Ind.   */
                 char cen_tract_00[6];    /* 2000 Census Tract        */
                 char cen_blk_00[4];      /* 2000 Census Block        */
                 char instruc_div[2];     /* Instructional Division   */
                 char filler07[2];        /* Filler                   */
                 char health_area[4];     /* Health Area              */
                 char sanit_recycle[3];   /* Recycling Sanit pick-up  */
                 char feature_type;       /* Feature Type Code        */
                 char iaei;               /* Interim Ass'tance Elig   */
                                          /* Indicator                */
                 char curve_flag;         /* Curve Flag               */
                 char police_boro_com;    /* Police Patrol Boro Commnd*/
                 char police_pre[3];      /* Police Precinct          */
                 char com_schl_dist[2];   /* Community School District*/
                                          /* Following 7 fields are   */
                                          /* used for Function 1E only*/
                 char ed[3];              /* Election District        */
                 char ad[2];              /* Assembly District        */
                 char sped_flag;          /* Split Elect District Flag*/
                 char congress_dist[2];   /* Congressional District   */
                 char state_sen_dist[2];  /* State Senatorial District*/
                 char civil_court[2];     /* Civil Court District     */
                 char civil_council[2];   /* City Council District    */
                 char filler06;
                 char segtypecode;        /* Segment Type Code        */
                 char sanit_dist[3];      /* Sanitation District      */
                 char sanit_sub_sect[2];  /* Sanit Collect Scheduling */
                                          /* Section and Subsection   */
                 char fire_divisn[2];     /* Fire Division            */
                 char fire_bat[2];        /* Fire Battalion           */
                 char fire_co_type;       /* Fire Company Type        */
                 char fire_co_nbr[3];     /* Fire Company Number      */
                 char sagr_flag;          /* Special Address Generated*/
                                          /* Record flag              */
                 char mh_ri_flag;         /* Marble Hill/Rikers Island*/
                                          /* Alternative Borough flag */
                 char scsd_flag;          /* Split Com School District*/
                                          /* flag                     */
```

WAC COPY File (continued)

```
                 char DCP_lgc[2];        /* DCP preferred LGC        */
                 char face_code[4];      /* LION Face Code           */
                 char seq_nbr[5];        /* LION Sequence Number     */
                 char cen_tract_90[6];   /* 1990 Census Tract        */
                 char filler09  [4];     /* Filler                   */
                 char dynam_blk[3];      /* Dynamic Block            */
                 char X_coord[7];        /* X coordinate             */
                 char Y_coord[7];        /* Y coordinate             */
                 char seg_len[5];        /* Segment Length in Feet   */
                 char sanit_reg_sched[5];/* Regularly Sanit pick-up  */
               } C_WA2_F1;

typedef struct { C_WA2_F1 c_wa2_f1;      /* First 200 Bytes          */
                 char seg_id[7];         /* Segment Identifier       */
                 char true_b7sc[8];      /* "true" Boro 7 Str code   */
                 char true_hni [6];      /* Underlying HNI           */
                 char filler01[79];      /* Filler - Future Use      */
               } C_WA2_F1L;

typedef struct { char lo_hse_nbr[6];     /* Low House nbr in HNI form*/
                 char filler01[3];
                 char hi_hse_nbr[6];     /* Hi House Nbr in HNI form */
                 char filler02[3];
                 char B5SC[6];           /* Boro & 5 digit Str Code  */
                 char lgc[2];            /* LGC of Street            */
                 char bld_id[7];         /* BIN of address range     */
                 char addr_type;         /* Address Type             */
                 char filler04;
                 char sos_ind;           /* Side of Street Indicator */
               } ADDR_RANGE;

typedef struct { char sanborn_boro;      /* Sanborn Borough Code     */
                 char sanborn_vol[3];    /* Sanborn Volume           */
                 char sanborn_page[4];   /* Sanborn Page             */
               } SANBORN;

typedef struct { char filler01[21];
                 char cont_parity_ind;   /* Continuous Parity Ind    */
                 char lo_hse_nbr[6];     /* Low House Number         */
                 char BBL[10];           /* Boro(len=1), Block(len=5)*/
                                         /* and Lot (len=4)          */
                 char tax_lot_ver_nbr;   /* Tax Lot Version Number   */
                 char RPAD_scc;          /* RPAD Self_Check Code(SCC)*/
                 char filler02;
                 char RPAD_lucc[2];      /* RPAD Land Use Class. Code*/
                 char corner[2];         /* Corner Code              */
                 char nbr_blds[2];       /* Nbr of buildings on lot  */
                 char nbr_str[2];        /* Nbr Street Frontages     */
                 char inter_flag;        /* Interior Lot Flag        */
                 char vacant_flag;       /* Vacant Lot Flag          */
                 char irreg_flag;        /* Irregularly-Shaped Lot Fl*/
                 char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                 char filler03;          /* Former Pseudo-Address Flg*/
                 char stroll_key[13];    /* Strolling key            */
                 char overflow_flag;     /* More than 21 Addresses   */
                 char res_internal_use;  /* Reserved for Internal Use*/
                 char bld_id[7];         /* Bld Identification Nbr   */
                                         /* (BIN) of Input Address of*/
                                         /* Existing Building, If any*/
                 char condo_flag;        /* Condominium Flag         */
                 char RPAD_cin[4];       /* RPAD Condo Id Number     */
                 char condo_lo_BBL[10];  /* Low BBL of Condo         */
                 char filler05;
                 char condo_bill_BBL[10];/* Condo Billing BBL        */
                 char filler06;
                 char condo_bill_BBL_scc;/* Condo Billing BBL        */
```

WAC COPY File (continued)

```
                            /* Self-Check Code          */ 00021003
          char condo_hi_BBL[10];  /* High BBL of Condo        */ 00021103
          char filler07;                                       00021203
          SANBORN fn1A_Sanborn;   /* Sanborn Information      */ 00021303
          char business_area[5];  /* Business Area            */ 00021403
          char co_op_nbr[4 ];     /* Co-op Number             */ 00021503
          char filler08[4 ];                                   00021711
          char tot_nbr_bldgs[4 ]; /* Actual Nbr Bldgs on lot  */ 00021803
          char tax_map_nbr[5 ];   /* Tax Map Nbr-Sect and Vol */ 00021911
          char filler09[04];                                   00022020
          char X_coord[7];        /* X coordinate-Annotation p*/ 00022120
          char Y_coord[7];        /* Y coordinate-Annotation p*/ 00022220
          char filler10[18];                                   00022320
          char nbr_addr[2];       /* Nbr of Addr Ranges on Lot*/ 00022403
          ADDR_RANGE addr_range[21]; /* Addr Range structure  */ 00022503
        } C_WA2_F1A;                                           00022603
                                                               00022703
typedef struct { char filler01[21];                            00022803
          char cont_parity_ind;   /* Continuous Parity Ind    */ 00022903
          char lo_hse_nbr[6];     /* Low House Number         */ 00023003
          char BBL[10];           /* Boro(len=1), Block(len=5)*/ 00023103
                                  /* and Lot (len=4)          */ 00023203
          char tax_lot_ver_nbr;   /* Tax Lot Version Number   */ 00023303
          char RPAD_scc;          /* RPAD Self_Check Code(SCC)*/ 00023403
          char filler02;                                       00023503
          char RPAD_lucc[2];      /* RPAD Land Use Class. Code*/ 00023603
          char corner[2];         /* Corner Code              */ 00023703
          char nbr_blds[2];       /* Nbr of buildings on lot  */ 00023803
          char nbr_str[2];        /* Nbr Street Frontages     */ 00023903
          char inter_flag;        /* Interior Lot Flag        */ 00024003
          char vacant_flag;       /* Vacant Lot Flag          */ 00024103
          char irreg_flag;        /* Irregularly-Shaped Lot Fl*/ 00024203
          char mh_ri_flag;        /* Marble Hill/Rikers Island*/ 00024303
          char filler03;          /* Former Pseudo-Address Flg*/ 00024403
          char stroll_key[13];    /* Strolling key            */ 00024503
          char filler04;                                       00024603
          char res_internal_use;  /* Reserved for Internal Use*/ 00024703
          char bld_id[7];         /* Bld Identification Nbr   */ 00024803
                                  /* (BIN) of Input Address of*/ 00024903
                                  /* Existing Building, If any*/ 00025003
          char condo_flag;        /* Condominium Flag         */ 00025103
          char RPAD_cin[4];       /* RPAD Condo Id Number     */ 00025203
          char condo_lo_BBL[10];  /* Low BBL of Condo         */ 00025303
          char filler05;                                       00025403
          char condo_bill_BBL[10];/* Condo Billing BBL        */ 00025503
          char filler06;                                       00025603
          char condo_bill_BBL_scc;/* Condo Billing BBL        */ 00025703
                                  /* Self-Check Code          */ 00025803
          char condo_hi_BBL[10];  /* High BBL of Condo        */ 00025903
          char filler07;                                       00026003
          SANBORN fn1A_Sanborn;   /* Sanborn Information      */ 00026103
          char business_area[5];  /* Business Area            */ 00026203
          char co_op_nbr[4 ];     /* Co-op number             */ 00026303
          char filler08[4 ];                                   00026411
          char tot_nbr_bldgs[4 ]; /* Actual Nbr Bldgs on lot  */ 00026503
          char tax_map_nbr[5 ];   /* Tax Map Nbr-Sect and Vol */ 00026611
          char filler09[04];                                   00026720
          char X_coord[7];        /* X coordinate-Annotation p*/ 00026820
          char Y_coord[7];        /* Y coordinate-Annotation p*/ 00026920
          char filler10[16];                                   00027020
          char nbr_bins[4];       /* Nbr of BINS on Lot       */ 00027103
          char bin_list[2500][7]; /* List of BINS on Lot      */ 00027203
        } C_WA2_F1AL;                                          00027303
                                                               00027403
typedef struct { char filler01[31];                            00027503
          char lgc[2][2];         /* Preferred LGCs           */ 00027603
```

WAC COPY File (continued)

```
                 char nbr_x_sts;          /* Number of Intersecting St*/
                 char x_sts[5][4];        /* PB5SCs of Intersection St*/
                 char compdir[1];         /* Compass Direction if 2   */
                                          /* lowest str codes cross   */
                                          /* exactly twice            */
                 char level_codes[10];    /* Level Codes of X Streets */
                 char instruc_div[2];     /* Instructional Division   */
                 char fire_sector[2];     /* Fire Sector              */
                 char fire_bat[2];        /* Fire Battalion           */
                 char fire_co_type;       /* Fire Company Type        */
                 char fire_co_nbr[3];     /* Fire Company Number      */
                 char com_dist[3];        /* Community District       */
                                          /* Pos 0 contains the Boro  */
                                          /* Code and Positions 1 & 2 */
                                          /* contain the district nbr */
                 char zip_code[5];        /* Zip code for st segment  */
                 char DOT_slca;           /* DOT St Lght Contractr Are*/
                 char cen_tract_00[6];    /* 2000 Census Tract        */
                 char filler03[3];
                 char health_area[4];     /* Health Area              */
                 char filler04[9];
                 char node_nbr[7 ];       /* Node Number              */
                 char X_coord[7];         /* X coordinate             */
                 char Y_coord[7];         /* Y coordinate             */
                 char filler04a[2];       /*                          */
                 char filler05[2];
                 char police_boro_com;    /* Police Patrol Boro Commnd*/
                 char police_pre[3];      /* Police Precinct          */
                 char com_schl_dist[2];   /* Community School District*/
                                          /* Following 7 fields are   */
                                          /* used forFunction 1E only */
                 char mh_ri_flag;         /* Marble Hill/Rikers Island*/
                 char cen_tract_90[6];    /* 1990 Census Tract        */
                 SANBORN fn2_Sanborn[2];  /* Sanborn Information      */
                 char filler06[38];
               } C_WA2_F2;

typedef struct { char filler01[21];
                 char dup_key_flag;       /* Duplicate Key Flag      */
                 char curve_flag;         /* Curve Flag              */
                 char loc_stat_seg;       /* Locational Status of Seg*/
                 char cnty_bnd_ind;       /* County Boundary Indicat */
                 char filler03[4];
                 char lgc[3][2];          /* Preferred LGCs           */
                 char lo_nbr_x_sts;       /* Nbr of cross sts at low  */
                                          /* house nbr end of street  */
                 char l_x_sts[5][4];      /* PB5SCs of lo end X sts   */
                 char hi_nbr_x_sts;       /* Number of X streets at lo*/
                                          /* house nbr end of street  */
                 char h_x_sts[5][4];      /* PB5SCs of low end X sts  */
                 char DOT_slca;           /* DOT St Lght Contractr Are*/
                 char x_street_reversal_flag; /* X St Reversal Flag  */
                 char l_com_dist[3];      /* Left Community District  */
                                          /* Position 0 contains the  */
                                          /* Boro Code and Pos 1 & 2  */
                                          /* contain the district nbr */
                 char r_com_dist[3];      /* Right Community District */
                                          /* Position 0 contains the  */
                                          /* Boro Code and Pos 1 & 2  */
                                          /* contain the district nbr */
                 char l_zip_code[5];      /* Left Zip code for st seg */
                 char r_zip_code[5];      /* Right Zip code for st seg*/
                 char filler07 [18];
                 char l_health_area[4];   /* Left Health Area         */
                 char r_health_area[4];   /* Right Health Area        */
                 char l_instruc_div[2];   /* Left Instructional Div   */
```

WAC COPY File (continued)

```
                 char r_instruc_div[2];  /* Right Instructional Div  */
                 char l_lo_hse_nbr[7];   /* Left Lo Hse nbr in Disp  */
                 char l_hi_hse_nbr[7];   /* Left Hi Hse Nbr in Disp  */
                 char r_lo_hse_nbr[7];   /* rght Lo Hse nbr in Disply*/
                 char r_hi_hse_nbr[7];   /* rght Hi Hse Nbr in Disply*/
                 char cont_par;          /* Continuous Parity Ind    */
                 char face_code[4];      /* LION Face Code           */
                 char seq_nbr[5];        /* LION Sequence Nbr        */
                 char genr_flag;         /* Generated Record Flag    */
                 char seg_len[3];        /* Segment Length in Feet   */
                 char seg_azm[3];        /* Segment Azimuth          */
                 char seg_orient;        /* Segment Orientation      */
                 char filler04a[2];      /*                          */
                 char filler04b[2];      /*                          */
                 char l_iaei;            /* Interim Assistance       */
                                         /* Eligibility Indicator for*/
                                         /* left side                */
                 char r_iaei;            /* Interim Assistance       */
                                         /* Eligibility Indicator for*/
                                         /* right side               */
                 char dog_leg;           /* Dog Leg Flag             */
                 char feature_type;      /* Feature Type Code        */
                 char l_police_boro_com; /* Lft Police Patrl Boro Com*/
                 char l_police_pre[3];   /* Left Police Precinct     */
                 char r_police_boro_com; /* Rght Police Patrl Boro Cm*/
                 char r_police_pre[3];   /* Right Police Precinct    */
                 char l_com_schl_dist[2];/* Lft Com School District  */
                 char r_com_schl_dist[2];/* Rght Com School District */
                 char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                                         /* Alternative Boro flag    */
                 char seg_id[7];         /* Segment Identifier       */
                 char segtypecode;       /* Segment Type Code        */
               } C_WA2_F3;

typedef struct { C_WA2_F3 c_wa2_f3;      /* First 200 Bytes          */
                 char l_cen_tract_90[6]; /* Left 1990 Census Tract   */
                 char filler01[4];       /* Filler                   */
                 char l_dynam_blk[3];    /* Left Dynamic Block       */
                 char r_cen_tract_90[6]; /* Right 1990 Census Tract  */
                 char filler03[4];       /* Filler                   */
                 char r_dynam_blk[3];    /* Right Dynamic Block      */
                 char l_fire_sector[2];  /* Left Fire Sector         */
                 char l_fire_bat[2];     /* Left Fire Battalion      */
                 char l_fire_co_type;    /* Left Fire Company Type   */
                 char l_fire_co_nbr[3];  /* Left Fire Company Nbr    */
                 char r_fire_sector[2];  /* Right Fire Sector        */
                 char r_fire_bat[2];     /* Right Fire Battalion     */
                 char r_fire_co_type;    /* Right Fire Company Type  */
                 char r_fire_co_nbr[3];  /* Right Fire Company Nbr   */
                 char l_cen_tract_00[6]; /* Left 2000 Census Tract   */
                 char l_cen_blk_00[4];   /* Left 2000 Census Block   */
                 char filler04;          /* Possible Census Blk Suff */
                 char r_cen_tract_00[6]; /* Right 2000 Census Tract  */
                 char r_cen_blk_00[4];   /* Right 2000 Census Block  */
                 char filler05;          /* Possible Census Blk Suff */
                 char filler02[36];
               } C_WA2_F3L;

typedef struct { char filler01[21];
                 char curve_flag;        /* curve flag               */
                 char segtypecode;       /* Segment Type Code        */
                 char loc_stat_seg;      /* Location Status of Seg.  */
                 char cnty_bnd_ind;      /* County Boundary Indicator*/
                 char filler0A[4];
                 char lgc[3][2];         /* Preferred LGCs           */
                 char lo_nbr_x_sts;      /* Nbr of cross sts at low  */
```

WAC COPY File (continued)

```
                             /* house nbr end of street  */
      char l_x_sts[5][4];    /* PB5SCs of lo end cross st*/
      char hi_nbr_x_sts;     /* Nbr of cross sts at low  */
                             /* house nbr end of street  */
      char h_x_sts[5][4];    /* PB5SCs of lo end X sts   */
      char com_dist[3];      /* Community District Pos 0 */
                             /* contains the Boro Code & */
                             /* Positions 1&2 contain the*/
                             /* district nbr             */
      char zip_code[5];      /* Zip code for street seg  */
      char DOT_slca;         /* DOT St Lght Contractr Are*/
      char filler02[7];
      char cen_tract_00[6];  /* 2000 Census Tract        */
      char cen_blk_00[4];    /* 2000 Census Block        */
      char filler04;         /* Possible Census Blk Suff */
      char health_area[4];   /* Health Area              */
      char x_street_reversal_flag; /* X St Reversal Flag  */
      char sos_ind;          /* Side of Street Indicator */
      char fire_sector[2];   /* Fire Sector              */
      char fire_bat[2];      /* Fire Battalion           */
      char fire_co_type;     /* Fire Company Type        */
      char fire_co_nbr[3];   /* Fire Company Nbr         */
      char seg_id[7];        /* Segment Identifier       */
      char lo_hse_nbr[7];    /* Low House nbr in Display */
      char hi_hse_nbr[7];    /* High House Nbr in Display*/
      char a_lo_hse_nbr[7];  /* Alt. Lo Hse nbr in Disply*/
      char a_hi_hse_nbr[7];  /* Alt.Hi Hse Nbr in Display*/
      char cont_par;         /* Continuous Parity Ind    */
      char face_code[4];     /* LION Face Code           */
      char seq_nbr[5];       /* LION Sequence Nbr        */
      char genr_flag;        /* Generated Record Flag    */
      char seg_len[3];       /* Segment Length in Feet   */
      char seg_azm[3];       /* Segment Azimuth          */
      char seg_orient;       /* Segment Orientation      */
      char instruc_div[2];   /* Instructional Division   */
      char iaei;             /* Interim Assistance       */
                             /* Eligibility Indicator    */
      char feature_type;     /* Feature Type Code        */
      char police_boro_com;  /* Police Patrol Boro Com.  */
      char police_pre[3];    /* Police Precinct          */
      char com_schl_dist[2]; /* Community School District*/
      char mh_ri_flag;       /* Marble Hill/Rikers Island*/
                             /* Alternative Boro flag    */
      char cen_tract_90[6];  /* 1990 Census Tract        */
      char filler03[4];      /* Filler                   */
      char dynam_blk[3];     /* Dynamic Block            */
      char filler06[5];
    } C_WA2_F3C;

typedef struct { char lo_x_PB5SC[4];     /* Lowest PB5SC at Intersect*/
      char lo2x_PB5SC[4];    /* 2nd Lowest PB5SC at Inter*/
      char len[3];           /* Len in ft from prev node */
      char gap_flag;         /* Gap Flag                 */
    } CROSS_STRS;

typedef struct { char filler01[21];
      char nbr_x_str[3];     /* Nbr of X sts in list     */
      CROSS_STRS cross_strs[350];/* Cross Street structure*/
    } C_WA2_F3S;
```

NATURAL LDAs (MSW)

GEOLW1 COPY File

```
*     USER PROGRAMS MUST RESET GEOLW1 BEFORE PRIMING WORKAREA 1
   1 GEOLW1                                    /* LRECL=200
*     THE FIELD W1NAT IS USED AS A PARAMETER TO CALL GEOSUPPORT
   2 W1NAT                         A          2
R  2 W1NAT
*  * * * *   INPUT FIELDS    * * * *  *
   3 GEO-WA1-IN-FUNCTION-CODE      A          2 /* BEGINNING OF FCT 1 LAYOUT
R  3 GEO-WA1-IN-FUNCTION-CODE
   4 GEO-WA1-IN-FUNCTION-1         A          1
   4 GEO-WA1-IN-FUNCTION-2         A          1
   2 GEO-WA1-IN-BORO               A          1
   2 GEO-WA1-IN-HOUSENUM           A         12 /* FOR FCT 5, INPUT HIGH HSE NUM
   2 GEO-WA1-IN-HOUSENUM-INTERNAL  A          6 /* FOR FCT 5, INPUT HIGH HSE NUM
   2 GEO-WA1-IN-STREET-1           A         32
   2 GEO-WA1-IN-STREET-2           A         32
   2 GEO-WA1-IN-STREET-3           A         32
   2 GEO-WA1-IN-COMPASS            A          1
   2 GEO-WA1-IN-COMPASS2           A          1
   2 GEO-WA1-IN-STREETCODE-1       P          6
   2 GEO-WA1-IN-STREETCODE-2       P          6
   2 GEO-WA1-IN-STREETCODE-3       P          6
   2 GEO-WA1-IN-ROADBED-REQ-SWITCH A          1
   2 GEO-WA1-IN-BORO-2             A          1
   2 GEO-WA1-IN-BORO-3             A          1
   2 GEO-WA1-IN-SNL                A          2
   2 GEO-WA1-IN-10SC-1             A         11
   2 GEO-WA1-IN-10SC-2             A         11
   2 GEO-WA1-IN-10SC-3             A         11
   2 GEO-WA1-IN-CUI                A          5 /* NOT IMPLEMENTED
   2 GEO-WA1-IN-BBL                A         10
R  2 GEO-WA1-IN-BBL
   3 GEO-WA1-IN-BBL-BORO           A          1
   3 GEO-WA1-IN-BLOCKNUM           A          5
   3 GEO-WA1-IN-LOTNUM             A          4
   2 FILLER-WA1-10                 A          1
   2 GEO-WA1-IN-BIN                A          7
   2 GEO-WA1-IN-COMPACT-NAME-FLAG  A          1 /* TO REQUEST THE COMPACT NAMES
*                                                  OPTION,
*                                                  MOVE 'C' TO THIS FIELD.
*
   2 GEO-WA1-IN-LONG-WORKAREA2-FLAG  A        1 /* TO REQUEST THE LONG WORKAREA
*                                                  OPTION,
*                                                  MOVE 'L' TO THIS FIELD.
*                                                  ONLY FCT 3 HAS THIS OPTION.
*
   2 GEO-WA1-IN-LOW-HOUSENUM       A         12
   2 GEO-WA1-IN-LOW-HSENUM-INTERNAL  A        6
   2 GEO-WA1-IN-NON-IBM-MAIN-FRAME A          1 /* NOT IMPLEMENTED
*                                                  FOR ANY APPLICATION RUNNING
*                                                  ON NON-IBM MAINFRAM,
*                                                  MOVE 'X' TO THIS FIELD.
*
   2 GEO-WA1-IN-1ABL-VERSION       A          1 /* FOR FCT 1A & BL, TO REQUEST
*                                                  THE STANDARD WA2 FORMAT, SET
*                                                  THIS FIELD TO 'S'. TO REQUEST
*                                                  THE LEGACY WA2 FORMAT, SET
*                                                  THIS FILED TO ' ' OR 'L'.
*
   2 GEO-WA1-IN-XSTREET-FLAG       A          1
   2 FILLER-WA1-100                A          4
*  * * * *   OUTPUT FIELDS   * * * *  *
   2 GEO-WA1-OUT-LOW-HOUSENUM      A         12
   2 GEO-WA1-OUT-BORONAME          A          9
   2 GEO-WA1-OUT-STREET-1          A         32
   2 GEO-WA1-OUT-STREET-2          A         32
```

GEOLW1 COPY File (continued)

```
    2 GEO-WA1-OUT-STREET-3               A          32
    2 GEO-WA1-OUT-HOUSENUM               A          12 /* HI HND
    2 GEO-WA1-OUT-HOUSENUM-INTERNAL      A           6
    2 FILLER-WA1-200                     A           7
    2 GEO-WA1-OUT-PB5SC-1                P           6 /* 4 BYTES
R   2 GEO-WA1-OUT-PB5SC-1
    3 GEO-WA1-OUT-PACKBORO-NOSIGN-1      N           1
    3 GEO-WA1-OUT-STREETCODE-1-KEY       P           5 /* 3 BYTES
    2 FILLER-WA1-210                     A           2
    2 GEO-WA1-OUT-PB5SC-2                P           6 /* 4 BYTES
R   2 GEO-WA1-OUT-PB5SC-2
    3 GEO-WA1-OUT-PACKBORO-NOSIGN-2      N           1
    3 GEO-WA1-OUT-STREETCODE-2-KEY       P           5 /* 3 BYTES
    2 FILLER-WA1-220                     A           2
    2 GEO-WA1-OUT-PB5SC-3                P           6 /* 4 BYTES
R   2 GEO-WA1-OUT-PB5SC-3
    3 GEO-WA1-OUT-PACKBORO-NOSIGN-3      N           1
    3 GEO-WA1-OUT-STREETCODE-3-KEY       P           5 /* 3 BYTES
    2 GEO-WA1-OUT-STREET-ATTR            A           1 (1:3) /* INTERNAL USE
    2 GEO-WA1-BROWSE                     A          40
    2 GEO-WA1-OUT-10SC-1                 A          11
    2 GEO-WA1-OUT-10SC-2                 A          11
    2 GEO-WA1-OUT-10SC-3                 A          11
    2 GEO-WA1-OUT-CUI                    A           5 /* NOT IMPLEMENTED
    2 GEO-WA1-OUT-BBL                    A          10
R   2 GEO-WA1-OUT-BBL
    3 GEO-WA1-OUT-BBL-BORO               A           1
    3 GEO-WA1-OUT-BLOCKNUM               A           5
    3 GEO-WA1-OUT-LOTNUM                 A           4
    2 FILLER-WA1-240                     A           1
    2 GEO-WA1-OUT-BIN                    A           7
    2 GEO-WA1-OUT-SND-ATTR               A           1 /* DCP/GSS USE
    2 GEO-WA1-OUT-REASON-CODE            A           1
    2 FILLER-WA1-400                     A           2 /* INTERNAL USE
    2 GEO-WA1-OUT-RETURN-CODE            A           2
R   2 GEO-WA1-OUT-RETURN-CODE
    3 GEO-WA1-OUT-RC-1                   A           1
    3 GEO-WA1-OUT-RC-2                   A           1
    2 GEO-WA1-OUT-ERROR-MESSAGE          A          80
    2 GEO-WA1-OUT-NUM-SIMILAR-NAMES      P           3
    2 GEO-WA1-OUT-SIMILAR-NAMES          A          32 (1:10)
```

GEOLW2 COPY File

```
  1 GEOLW2
*   THE FIELD W2NAT IS USED AS A PARAMETER TO CALL GEOSUPPORT FOR ALL
*       FUNCTIONS THAT ARE REDEFINED ON GEOLW2
  2 W2NAT                            A   21
R 2 W2NAT
* *  BEGINNING OF FUNCTION 1 LAYOUT  * **** *******
  3 GEO-WA2-FN1-ACCESS-KEY           A   21
  2 GEO-WA2-FN1-CONT-PARITY          A    1
  2 GEO-WA2-FN1-LOW-HOUSENUM-INT     A    6
R 2 GEO-WA2-FN1-LOW-HOUSENUM-INT
  3 GEO-WA2-FN1-LOW-HOUSENUM         A    5
  3 GEO-WA2-FN1-LOW-HOUSENUMSFX      A    1
  2 GEO-WA2-FN1-HI-HOUSENUM-INT      A    6
R 2 GEO-WA2-FN1-HI-HOUSENUM-INT
  3 GEO-WA2-FN1-HI-HOUSENUM          A    5
  3 GEO-WA2-FN1-HI-HOUSENUMSFX       A    1
  2 ALX                              A    1 /*ALLEYS INTERSECT SEGMENT
  2 GEO-WA2-FN1-NUM-X-ST-LOW-END     N    1
  2 GEO-WA2-FN1-LOW-PBSC             P    6 (1:5) /* 4 BYTES X 5 = 20
  2 GEO-WA2-FN1-NUM-X-ST-HI-END      N    1
  2 GEO-WA2-FN1-HI-PBSC              P    6 (1:5)
  2 GEO-WA2-FN1-COMDIST              A    3
R 2 GEO-WA2-FN1-COMDIST
  3 GEO-WA2-FN1-COMDIST-BORO         A    1
  3 GEO-WA2-FN1-COMDIST-NUM          A    2
  2 GEO-WA2-FN1-ZIP                  A    5
  2 GEO-WA2-FN1-SLA                  A    1
  2 GEO-WA2-FN1-HCD                  A    2
  2 GEO-WA2-FN1-SOS                  A    1
  2 GEO-WA2-FN1-CONT-PARITY-IND      A    1
  2 GEO-WA2-FN1-2000-CENSUSTRACT     A    6
  2 GEO-WA2-FN1-2000-CENSUSBLOCK     A    4
  2 GEO-WA2-FN1-INSTRUCT-REG         A    2
  2 FILLER-WA2-260                   A    2
  2 GEO-WA2-FN1-HEALTHAREA           A    4
  2 GEO-WA2-FN1-SANI-REC             A    3
  2 FILLER-WA2-230                   A    1
  2 GEO-WA2-FN1-RESDCP               A    1 /* RESERVED FOR DCP/GSS USE
  2 GEO-WA2-FN1-CURVE-FLAG           A    1
  2 GEO-WA2-FN1-POLICEDIST           A    4
R 2 GEO-WA2-FN1-POLICEDIST
  3 GEO-WA2-FN1-POL-PATR-BORO-CMD    A    1
  3 GEO-WA2-FN1-POL-PRECINCT         A    3
  2 GEO-WA2-FN1-SCHOOLDIST           A    2
  2 FILLER-WA2-250                   A   15
  2 GEO-WA2-FN1-SEGMENT-TYPE         A    1
  2 GEO-WA2-FN1-SANIDIST             A    3
R 2 GEO-WA2-FN1-SANIDIST
  3 GEO-WA2-FN1-SANIDIST-BORO        A    1
  3 GEO-WA2-FN1-SANIDIST-NUM         A    2
  2 GEO-WA2-FN1-SANITATION-SUBSEC    A    2
  2 GEO-WA2-FN1-FIRESEC              A    2 /* FIRE DIVISION
  2 GEO-WA2-FN1-FIREBAT              A    2
  2 GEO-WA2-FN1-FIRECO               A    4
R 2 GEO-WA2-FN1-FIRECO
  3 GEO-WA2-FN1-FIRECO-TYPE          A    1
  3 GEO-WA2-FN1-FIRECO-NUM           A    3
  2 GEO-WA2-FN1-SPECIAL-ADDR-FLAG    A    1
  2 GEO-WA2-FN1-MARBLE-RIKERS-FLAG   A    1
  2 GEO-WA2-FN1-SPLIT-SCHOOL-FLAG    A    1
  2 GEO-WA2-FN1-PREFERRED-LGC        A    2
  2 GEO-WA2-FN1-LIONFACECODE         A    4
  2 GEO-WA2-FN1-LIONSEQ              A    5
  2 GEO-WA2-FN1-1990-CENSUSTRACT     A    6
  2 FILLER-WA2-260B                  A    4
```

GEOLW2 COPY File (continued)

```
  2 GEO-WA2-FN1-DYN-BLOCK            A    3
  2 GEO-WA2-FN1-XCOORD               A    7
  2 GEO-WA2-FN1-YCOORD               A    7
  2 GEO-WA2-FN1-SEGMENTLENGTH        A    5
  2 GEO-WA2-FN1-SANI-REG             A    5
* *  END OF FUNCTION 1 LAYOUT        * **** *******
* - ------------------------------- - ---- -------------------------------
* *  BEGINNING OF FUNCTION 2 LAYOUT  * **** *******
R 1 GEOLW2
  2 GEO-WA2-FN2-ACCESS-KEY           A   21 /*FCT 2,2C SHARE SAME WA2 LAYOUT
  2 GEO-WA2-FN2-DUPINTERFLAG         A    1 /*
  2 FILLER-WA2-270                   A    9
  2 GEO-WA2-FN2-PREFERRED-LGC1       A    2
  2 GEO-WA2-FN2-PREFERRED-LGC2       A    2
  2 GEO-WA2-FN2-NUM-OF-INTERSECTS    N    1
  2 GEO-WA2-FN2-INTERSECT-PBSC       P    6 (1:5)
  2 GEO-WA2-FN2-COMPDIR              A    1
  2 GEO-WA2-FN2-LEVEL-CODES-TBL      A   10
R 2 GEO-WA2-FN2-LEVEL-CODES-TBL
  3 GEO-WA2-FN2-LEVEL-CODES          A    1 (5,2)
  2 GEO-WA2-FN2-INSTRUCT-REG         A    2
  2 GEO-WA2-FN2-FIRESEC              A    2
  2 GEO-WA2-FN2-FIREBAT              A    2
  2 GEO-WA2-FN2-FIRECO               A    4
R 2 GEO-WA2-FN2-FIRECO
  3 GEO-WA2-FN2-FIRECO-TYPE          A    1
  3 GEO-WA2-FN2-FIRECO-NUM           A    3
  2 GEO-WA2-FN2-COMDIST              A    3
R 2 GEO-WA2-FN2-COMDIST
  3 GEO-WA2-FN2-COMDIST-BORO         A    1
  3 GEO-WA2-FN2-COMDIST-NUM          A    2
  2 GEO-WA2-FN2-ZIP                  A    5
  2 GEO-WA2-FN2-SLA                  A    1
  2 GEO-WA2-FN2-2000-CENSUSTRACT     A    6
  2 FILLER-WA2-290                   A    3
  2 GEO-WA2-FN2-HEALTHAREA           A    4
  2 FILLER-WA2-300                   A    9
  2 GEO-WA2-FN2-LIONNODENUM          A    7
  2 GEO-WA2-FN2-XCOORD               A    7
  2 GEO-WA2-FN2-YCOORD               A    7
  2 FILLER-WA2-320                   A    4
  2 GEO-WA2-FN2-POLICEDIST           A    4
R 2 GEO-WA2-FN2-POLICEDIST
  3 GEO-WA2-FN2-POL-PATR-BORO-CMD    A    1
  3 GEO-WA2-FN2-POL-PRECINCT         A    3
  2 GEO-WA2-FN2-SCHOOLDIST           A    2
  2 GEO-WA2-FN2-MARBLE-RIKERS-FLAG   A    1
  2 GEO-WA2-FN2-1990-CENSUSTRACT     A    6
  2 GEO-WA2-FN2-SANBORN1-BVOLPAGE    A    8
R 2 GEO-WA2-FN2-SANBORN1-BVOLPAGE
  3 GEO-WA2-FN2-SANBORN1-BORO        A    1
  3 GEO-WA2-FN2-SANBORN1-VOLPAGE     A    7
R 3 GEO-WA2-FN2-SANBORN1-VOLPAGE
  4 GEO-WA2-FN2-SANBORN1-VOL-NUM     A    3
  4 GEO-WA2-FN2-SANBORN1-PAGE-NUM    A    4
  2 GEO-WA2-FN2-SANBORN2-BVOLPAGE    A    8
R 2 GEO-WA2-FN2-SANBORN2-BVOLPAGE
  3 GEO-WA2-FN2-SANBORN2-BORO        A    1
  3 GEO-WA2-FN2-SANBORN2-VOLPAGE     A    7
R 3 GEO-WA2-FN2-SANBORN2-VOLPAGE
  4 GEO-WA2-FN2-SANBORN2-VOL-NUM     A    3
  4 GEO-WA2-FN2-SANBORN2-PAGE-NUM    A    4
  2 FILLER-WA2-330                   A   38
* *  END OF FUNCTION 2 LAYOUT        * **** *******
* - ------------------------------- - ---- -------------------------------
* *  BEGINNING OF FUNCTION 3 LAYOUT  * **** *******
```

GEOLW2 COPY File (continued)

```
R 1 GEOLW2
  2 GEO-WA2-FN3-ACCESS-KEY            A   21
  2 GEO-WA2-FN3-DUP-KEY-FLAG          A    1 /* OR FN3 CONTI PARITY
  2 GEO-WA2-FN3-CURVE-FLAG            A    1
  2 GEO-WA2-FN3-LOCATION-STATUS       A    1
  2 GEO-WA2-FN3-COUNTY-BOUNDARY       A    1
  2 FILLER-WA2-340                    A    4
  2 GEO-WA2-FN3-PREFERRED-LGC1        A    2
  2 GEO-WA2-FN3-PREFERRED-LGC2        A    2
  2 GEO-WA2-FN3-PREFERRED-LGC3        A    2
  2 GEO-WA2-FN3-NUM-X-ST-LOW-END      N    1
  2 GEO-WA2-FN3-LOW-PBSC              P    6 (1:5)
  2 GEO-WA2-FN3-NUM-X-ST-HI-END       N    1
  2 GEO-WA2-FN3-HI-PBSC               P    6 (1:5)
  2 GEO-WA2-FN3-SLA                   A    1
  2 GEO-WA2-FN3-REVERSALFLAG          A    1
  2 GEO-WA2-FN3-LEFT-COMDIST          A    3
R 2 GEO-WA2-FN3-LEFT-COMDIST
  3 GEO-WA2-FN3-LEFT-COMDIST-BORO     A    1
  3 GEO-WA2-FN3-LEFT-COMDIST-NUM      A    2
  2 GEO-WA2-FN3-RIGHT-COMDIST         A    3
R 2 GEO-WA2-FN3-RIGHT-COMDIST
  3 GEO-WA2-FN3-RIGHT-COMDIST-BORO    A    1
  3 GEO-WA2-FN3-RIGHT-COMDIST-NUM     A    2
  2 GEO-WA2-FN3-LEFT-ZIP              A    5
  2 GEO-WA2-FN3-RIGHT-ZIP             A    5
  2 FILLER-WA2-350A                   A   18
  2 GEO-WA2-FN3-LEFT-HEALTHAREA       A    4
  2 GEO-WA2-FN3-RIGHT-HEALTHAREA      A    4
  2 GEO-WA2-FN3-LEFT-INSTRUCT-REG     A    2
  2 GEO-WA2-FN3-RIGHT-INSTRUCT-REG    A    2
  2 GEO-WA2-FN3-LEFT-LOW-HOUSENUM     A    7
  2 GEO-WA2-FN3-LEFT-HI-HOUSENUM      A    7
  2 GEO-WA2-FN3-RIGHT-LOW-HOUSENUM    A    7
  2 GEO-WA2-FN3-RIGHT-HI-HOUSENUM     A    7
  2 GEO-WA2-FN3-CONT-PARITY-IND       A    1
  2 GEO-WA2-FN3-LIONFACECODE          A    4
  2 GEO-WA2-FN3-LIONSEQ               A    5
  2 GEO-WA2-FN3-GENRECFLAG            A    1
  2 GEO-WA2-FN3-SEGMENTLENGTH         P    5
  2 GEO-WA2-FN3-SEGMENTSLOPE          A    3
  2 GEO-WA2-FN3-SEGMENTORIENT         A    1
  2 FILLER-WA2-355                    A    4
  2 GEO-WA2-FN3-RESSDCP               A    2 /* RESERVED FOR DCP/GSS USE
  2 GEO-WA2-FN3-DOG-LEG               A    1
  2 GEO-WA2-FN3-FEATURE-TYPE          A    1
  2 GEO-WA2-FN3-LEFT-POLICEDIST       A    4
R 2 GEO-WA2-FN3-LEFT-POLICEDIST
  3 GEO-WA2-FN3-L-POL-PATR-BORO-CMD   A    1
  3 GEO-WA2-FN3-LEFT-POL-PRECINCT     A    3
  2 GEO-WA2-FN3-RIGHT-POLICEDIST      A    4
R 2 GEO-WA2-FN3-RIGHT-POLICEDIST
  3 GEO-WA2-FN3-R-POL-PATR-BORO-CMD   A    1
  3 GEO-WA2-FN3-RIGHT-POL-PRECINCT    A    3
  2 GEO-WA2-FN3-LEFT-SCHOOLDIST       A    2
  2 GEO-WA2-FN3-RIGHT-SCHOOLDIST      A    2
  2 GEO-WA2-FN3-MARBLE-RIKERS-FLAG    A    1
  2 GEO-WA2-FN3-SEGMENT-ID            A    7
  2 GEO-WA2-FN3-SEGMENT-TYPE          A    1
* *  END OF FUNCTION 3 LAYOUT          **** ********
* - ------------------------------- - ---- ------------------------------
* *  BEGINNING OF FUNCTION 3C LAYOUT    **** ********
R 1 GEOLW2
  2 GEO-WA2-FN3C-ACCESS-KEY           A   21
  2 GEO-WA2-FN3C-CURVE-FLAG           A    1
  2 GEO-WA2-FN3C-SEGMENT-TYPE         A    1
```

GEOLW2 COPY File (continued)

```
  2 GEO-WA2-FN3C-LOCATION-STATUS      A    1
  2 GEO-WA2-FN3C-COUNTY-BOUNDARY      A    1
  2 FILLER-WA2-380                    A    4
  2 GEO-WA2-FN3C-PREFERRED-LGC1       A    2
  2 GEO-WA2-FN3C-PREFERRED-LGC2       A    2
  2 GEO-WA2-FN3C-PREFERRED-LGC3       A    2
  2 GEO-WA2-FN3C-NUM-X-ST-LOW-END     N    1
  2 GEO-WA2-FN3C-LOW-PBSC             P    6 (1:5)
  2 GEO-WA2-FN3C-NUM-X-ST-HI-END      N    1
  2 GEO-WA2-FN3C-HI-PBSC              P    6 (1:5)
  2 GEO-WA2-FN3C-COMDIST              A    3
R 2 GEO-WA2-FN3C-COMDIST
  3 GEO-WA2-FN3C-COMDIST-BORO         A    1
  3 GEO-WA2-FN3C-COMDIST-NUM          A    2
  2 GEO-WA2-FN3C-ZIP                  A    5
  2 GEO-WA2-FN3C-SLA                  A    1
  2 FILLER-WA2-390                    A    7
  2 GEO-WA2-FN3C-2000-CENSUSTRACT     A    6
  2 GEO-WA2-FN3C-2000-CENSUSBLOCK     A    4
  2 FILLER-WA2-390B-RES-DCP           A    1
  2 GEO-WA2-FN3C-HEALTHAREA           A    4
  2 GEO-WA2-FN3C-REVERSALFLAG         A    1
  2 GEO-WA2-FN3C-SOS                  A    1
  2 GEO-WA2-FN3C-FIRESEC              A    2
  2 GEO-WA2-FN3C-FIREBAT              A    2
  2 GEO-WA2-FN3C-FIRECO               A    4
R 2 GEO-WA2-FN3C-FIRECO
  3 GEO-WA2-FN3C-FIRECO-TYPE          A    1
  3 GEO-WA2-FN3C-FIRECO-NUM           A    3
  2 GEO-WA2-FN3C-SEGMENT-ID           A    7
  2 GEO-WA2-FN3C-LOW-HOUSENUM         A    7
  2 GEO-WA2-FN3C-HI-HOUSENUM          A    7
  2 GEO-WA2-FN3C-LOW-HOUSENUM2        A    7 /* HOUSENUM2 ONLY PRESENT IF ODD
  2 GEO-WA2-FN3C-HI-HOUSENUM2         A    7 /* ODD & EVEN ARE ON SOS  ME SOS
  2 GEO-WA2-FN3C-CONT-PARITY-IND      A    1
  2 GEO-WA2-FN3C-LIONFACECODE         A    4
  2 GEO-WA2-FN3C-LIONSEQ              A    5
  2 GEO-WA2-FN3C-GENRECFLAG           A    1
  2 GEO-WA2-FN3C-SEGMENTLENGTH        P    5
  2 GEO-WA2-FN3C-SEGMENTSLOPE         A    3
  2 GEO-WA2-FN3C-SEGMENTORIENT        A    1
  2 FILLER-WA2-408                    A    2
  2 GEO-WA2-FN3C-RESDCP               A    1 /* RESERVED FOR DCP/GSS USE
  2 FILLER-WA2-410                    A    1
  2 GEO-WA2-FN3C-POLICEDIST           A    4
R 2 GEO-WA2-FN3C-POLICEDIST
  3 GEO-WA2-FN3C-POL-PATR-BORO-CMD    A    1
  3 GEO-WA2-FN3C-POL-PRECINCT         A    3
  2 GEO-WA2-FN3C-SCHOOLDIST           A    2
  2 GEO-WA2-FN3C-MARBLE-RIKERS-FLAG   A    1
  2 GEO-WA2-FN3C-1990-CENSUSTRACT     A    6
  2 FILLER-WA2-410B                   A    4
  2 GEO-WA2-FN3C-DYN-BLOCK            A    3
  2 FILLER-WA2-420                    A    5
* *  END OF FUNCTION 3C LAYOUT       * **** *******
* - ------------------------------- - ---- -------------------------------
* *  BEGINNING OF FUNCTION 1E LAYOUT * **** *******
R 1 GEOLW2
  2 GEO-WA2-FN1E-ACCESS-KEY           A   21
  2 GEO-WA2-FN1E-CONT-PARITY          A    1
  2 GEO-WA2-FN1E-LOW-HOUSENUM-INT     A    6
R 2 GEO-WA2-FN1E-LOW-HOUSENUM-INT
  3 GEO-WA2-FN1E-LOW-HOUSENUM         A    5
  3 GEO-WA2-FN1E-LOW-HOUSENUMSFX      A    1 /* NOT IMPLEMENTED
  2 GEO-WA2-FN1E-HI-HOUSENUM-INT      A    6
R 2 GEO-WA2-FN1E-HI-HOUSENUM-INT
```

GEOLW2 COPY File (continued)

```
  3 GEO-WA2-FN1E-HI-HOUSENUM          A     5
  3 GEO-WA2-FN1E-HI-HOUSENUMSFX       A     1 /* NOT IMPLEMENTED
  2 FILLER-WA2-435                    A     1
  2 GEO-WA2-FN1E-NUM-X-ST-LOW-END     N     1
  2 GEO-WA2-FN1E-LOW-PBSC             P     6 (1:5)
  2 GEO-WA2-FN1E-NUM-X-ST-HI-END      N     1
  2 GEO-WA2-FN1E-HI-PBSC              P     6 (1:5)
  2 GEO-WA2-FN1E-COMDIST              A     3
R 2 GEO-WA2-FN1E-COMDIST
  3 GEO-WA2-FN1E-COMDIST-BORO         A     1
  3 GEO-WA2-FN1E-COMDIST-NUM          A     2
  2 GEO-WA2-FN1E-ZIP                  A     5
  2 GEO-WA2-FN1E-SLA                  A     1
  2 GEO-WA2-FN1E-HCD                  A     2
  2 GEO-WA2-FN1E-SOS                  A     1
  2 GEO-WA2-FN1E-CONT-PARITY-IND      A     1
  2 GEO-WA2-FN1E-2000-CENSUSTRACT     A     6
  2 GEO-WA2-FN1E-2000-CENSUSBLOCK     A     4
  2 GEO-WA2-FN1E-INSTRUCT-DIV         A     2
  2 FILLER-WA2-440                    A     2
  2 GEO-WA2-FN1E-HEALTHAREA           A     4
  2 GEO-WA2-FN1E-SANI-REC             A     3
  2 FILLER-WA2-450                    A     1
  2 GEO-WA2-FN1E-RESDCP               A     1 /* RESERVED FOR DCP/GSS USE
  2 GEO-WA2-FN1E-CURVE-FLAG           A     1
  2 GEO-WA2-FN1E-POLICEDIST           A     4
R 2 GEO-WA2-FN1E-POLICEDIST
  3 GEO-WA2-FN1E-POL-PATR-BORO-CMD    A     1
  3 GEO-WA2-FN1E-POL-PRECINCT         A     3
  2 GEO-WA2-FN1E-SCHOOLDIST           A     2
  2 GEO-WA2-FN1E-ELECTDIST            A     3
  2 GEO-WA2-FN1E-ASSEMDIST            A     2
  2 GEO-WA2-FN1E-SPLIT-ED-FLAG        A     1
  2 GEO-WA2-FN1E-CONGDIST             A     2
  2 GEO-WA2-FN1E-SENATEDIST           A     2
  2 GEO-WA2-FN1E-COURTDIST            A     2
  2 GEO-WA2-FN1E-COUNCILDIST          A     2
  2 FILLER-WA2-470                    A     2
  2 GEO-WA2-FN1E-SANIDIST             A     3
R 2 GEO-WA2-FN1E-SANIDIST
  3 GEO-WA2-FN1E-SANIDIST-BORO        A     1
  3 GEO-WA2-FN1E-SANIDIST-NUM         A     2
  2 GEO-WA2-FN1E-SANITATION-SUBSEC    A     2
  2 GEO-WA2-FN1E-FIRESEC              A     2
  2 GEO-WA2-FN1E-FIREBAT              A     2
  2 GEO-WA2-FN1E-FIRECO               A     4
R 2 GEO-WA2-FN1E-FIRECO
  3 GEO-WA2-FN1E-FIRECO-TYPE          A     1
  3 GEO-WA2-FN1E-FIRECO-NUM           A     3
  2 GEO-WA2-FN1E-SPECIAL-ADDR-FLAG    A     1
  2 GEO-WA2-FN1E-MARBLE-RIKERS-FLAG   A     1
  2 GEO-WA2-FN1E-SPLIT-SCHOOL-FLAG    A     1
  2 GEO-WA2-FN1E-PREFERRED-LGC        A     2
  2 GEO-WA2-FN1E-LIONFACECODE         A     4
  2 GEO-WA2-FN1E-LIONSEQ              A     5
  2 GEO-WA2-FN1E-1990-CENSUSTRACT     A     6
  2 FILLER-WA2-480B                   A     4
  2 GEO-WA2-FN1E-DYN-BLOCK            A     3
  2 GEO-WA2-FN1E-XCOORD               A     7
  2 GEO-WA2-FN1E-YCOORD               A     7
  2 GEO-WA2-FN1E-SEGMENTLENGTH        A     5
  2 GEO-WA2-FN1E-SANI-REG             A     5
* *  END OF FUNCTION 1E LAYOUT        * **** *******
* - ------------------------------- - ---- -------------------------------
* *  BEGINNING OF FUNCTION 5 LAYOUT   * **** *******
R 1 GEOLW2
```

GEOLW2 COPY File (continued)

```
  2 GEO-WA2-FN5-ADDR-MATCHING-KEY    A   28
  2 FILLER-WA2-490                   A  172
* *  END OF FUNCTION 5 LAYOUT        * **** *******
* - ------------------------------- - ---- --------------------------------
```

GEOLW2L COPY File

```
  1 GEOLW2L
* * THE FIELD W2NATL IS USED AS A PARAMETER TO CALL GEOSUPPORT
  2 W2NATL                          A   21
R 2 W2NATL
* * BEGINNING OF FUNCTION 1 LONG WORKAREA LAYOUT ******
  3 GEO-WA2-1L-ACCESS-KEY           A   21
  2 GEO-WA2-1L-CONT-PARITY          A    1
  2 GEO-WA2-1L-LOW-HOUSENUM-INT     A    6
R 2 GEO-WA2-1L-LOW-HOUSENUM-INT
  3 GEO-WA2-1L-LOW-HOUSENUM         A    5
  3 GEO-WA2-1L-LOW-HOUSENUMSFX      A    1
  2 GEO-WA2-1L-HI-HOUSENUM-INT      A    6
R 2 GEO-WA2-1L-HI-HOUSENUM-INT
  3 GEO-WA2-1L-HI-HOUSENUM          A    5
  3 GEO-WA2-1L-HI-HOUSENUMSFX       A    1
  2 FILLER-WA2-215                  A    1
  2 GEO-WA2-1L-NUM-X-ST-LOW-END     N    1
  2 GEO-WA2-1L-LOW-PBSC             P    6 (1:5)
  2 GEO-WA2-1L-NUM-X-ST-HI-END      N    1
  2 GEO-WA2-1L-HI-PBSC              P    6 (1:5)
  2 GEO-WA2-1L-COMDIST              A    3
R 2 GEO-WA2-1L-COMDIST
  3 GEO-WA2-1L-COMDIST-BORO         A    1
  3 GEO-WA2-1L-COMDIST-NUM          A    2
  2 GEO-WA2-1L-ZIP                  A    5
  2 GEO-WA2-1L-SLA                  A    1
  2 GEO-WA2-1L-HCD                  A    2
  2 GEO-WA2-1L-SOS                  A    1
  2 GEO-WA2-1L-CONT-PARITY-IND      A    1
  2 GEO-WA2-1L-2000-CENSUSTRACT     A    6
  2 GEO-WA2-1L-2000-CENSUSBLOCK     A    4
  2 GEO-WA2-1L-INSTRUCT-REG         A    2
  2 FILLER-1L-260-RESDCP            A    1 /* FILLER OF 2 ?
  2 FILLER-1L-260                   A    1
  2 GEO-WA2-1L-HEALTHAREA           A    4
  2 GEO-WA2-1L-SANI-REC             A    3
  2 FILLER-WA2-230                  A    1
  2 GEO-WA2-1L-RESDCP               A    1 /* RESERVED FOR DCP/GSS USE
  2 GEO-WA2-1L-CURVE-FLAG           A    1
  2 GEO-WA2-1L-POLICEDIST           A    4
R 2 GEO-WA2-1L-POLICEDIST
  3 GEO-WA2-1L-POL-PATR-BORO-CMD    A    1
  3 GEO-WA2-1L-POL-PRECINCT         A    3
  2 GEO-WA2-1L-SCHOOLDIST           A    2
  2 FILLER-WA2-250                  A   16 /* 1E POL DIST
  2 GEO-WA2-1L-SANIDIST             A    3
R 2 GEO-WA2-1L-SANIDIST
  3 GEO-WA2-1L-SANIDIST-BORO        A    1
  3 GEO-WA2-1L-SANIDIST-NUM         A    2
  2 GEO-WA2-1L-SANITATION-SUBSEC    A    2
  2 GEO-WA2-1L-FIRESEC              A    2 /* FIRE DIVISION
  2 GEO-WA2-1L-FIREBAT              A    2
  2 GEO-WA2-1L-FIRECO               A    4
R 2 GEO-WA2-1L-FIRECO
  3 GEO-WA2-1L-FIRECO-TYPE          A    1
  3 GEO-WA2-1L-FIRECO-NUM           A    3
  2 GEO-WA2-1L-SPECIAL-ADDR-FLAG    A    1
  2 GEO-WA2-1L-MARBLE-RIKERS-FLAG   A    1
  2 GEO-WA2-1L-SPLIT-SCHOOL-FLAG    A    1
  2 GEO-WA2-1L-PREFERRED-LGC        A    2
  2 GEO-WA2-1L-LIONFACECODE         A    4
  2 GEO-WA2-1L-LIONSEQ              A    5
  2 GEO-WA2-1L-1990-CENSUSTRACT     A    6
  2 FILLER-WA2-260B                 A    4
  2 GEO-WA2-1L-DYN-BLOCK            A    3
```

GEOLW2L COPY File (continued)

```
  2 GEO-WA2-1L-XCOORD              A    7
  2 GEO-WA2-1L-YCOORD              A    7
  2 GEO-WA2-1L-SEGMENTLENGTH       A    5
  2 GEO-WA2-1L-SANI-REG            A    5
  2 GEO-WA2-1L-SEGMENT-ID          A    7
  2 GEO-WA2-1L-TRUE-B7SC           A    8
  2 GEO-WA2-1L-UNDER-HN-INT        A    6
  2 FILLER-WA2-260C                A   79
* * END OF FUNCTION 1 LONG WORKAREA LAYOUT ******
* - -------------------------------- - ---- --------------------------------
* * BEGINNING OF FUNCTION 1E LONG WORKAREA LAYOUT ******
R 1 GEOLW2L
  2 GEO-WA2-1EL-ACCESS-KEY         A   21
  2 GEO-WA2-1EL-CONT-PARITY        A    1
  2 GEO-WA2-1EL-LOW-HOUSENUM-INT   A    6
R 2 GEO-WA2-1EL-LOW-HOUSENUM-INT
  3 GEO-WA2-1EL-LOW-HOUSENUM       A    5
  3 GEO-WA2-1EL-LOW-HOUSENUMSFX    A    1 /* NOT IMPLEMENTED
  2 GEO-WA2-1EL-HI-HOUSENUM-INT    A    6
R 2 GEO-WA2-1EL-HI-HOUSENUM-INT
  3 GEO-WA2-1EL-HI-HOUSENUM        A    5
  3 GEO-WA2-1EL-HI-HOUSENUMSFX     A    1 /* NOT IMPLEMENTED
  2 FILLER-WA2-435                 A    1
  2 GEO-WA2-1EL-NUM-X-ST-LOW-END   N    1
  2 GEO-WA2-1EL-LOW-PBSC           P    6 (1:5)
  2 GEO-WA2-1EL-NUM-X-ST-HI-END    N    1
  2 GEO-WA2-1EL-HI-PBSC            P    6 (1:5)
  2 GEO-WA2-1EL-COMDIST            A    3
R 2 GEO-WA2-1EL-COMDIST
  3 GEO-WA2-1EL-COMDIST-BORO       A    1
  3 GEO-WA2-1EL-COMDIST-NUM        A    2
  2 GEO-WA2-1EL-ZIP                A    5
  2 GEO-WA2-1EL-SLA                A    1
  2 GEO-WA2-1EL-HCD                A    2
  2 GEO-WA2-1EL-SOS                A    1
  2 GEO-WA2-1EL-CONT-PARITY-IND    A    1
  2 GEO-WA2-1EL-2000-CENSUSTRACT   A    6
  2 GEO-WA2-1EL-2000-CENSUSBLOCK   A    4
  2 GEO-WA2-1EL-INSTRUCT-REG       A    2
  2 FILLER-WA2-440-RES-DCP         A    1 /* FILLER OF 2 ?
  2 FILLER-WA2-440C                A    1
  2 GEO-WA2-1EL-HEALTHAREA         A    4
  2 GEO-WA2-1EL-SANI-REC           A    3
  2 GEO-WA2-1EL-FEATURE-TYPE       A    1
  2 GEO-WA2-1EL-RESDCP             A    1 /* RESERVED FOR DCP/GSS USE
  2 GEO-WA2-1EL-CURVE-FLAG         A    1
  2 GEO-WA2-1EL-POLICEDIST         A    4
R 2 GEO-WA2-1EL-POLICEDIST
  3 GEO-WA2-1EL-POL-PATR-BORO-CMD  A    1
  3 GEO-WA2-1EL-POL-PRECINCT       A    3
  2 GEO-WA2-1EL-SCHOOLDIST         A    2
  2 GEO-WA2-1EL-ELECTDIST          A    3
  2 GEO-WA2-1EL-ASSEMDIST          A    2
  2 GEO-WA2-1EL-SPLIT-ED-FLAG      A    1
  2 GEO-WA2-1EL-CONGDIST           A    2
  2 GEO-WA2-1EL-SENATEDIST         A    2
  2 GEO-WA2-1EL-COURTDIST          A    2
  2 GEO-WA2-1EL-COUNCILDIST        A    2
  2 FILLER-WA2-470                 A    2
  2 GEO-WA2-1EL-SANIDIST           A    3
R 2 GEO-WA2-1EL-SANIDIST
  3 GEO-WA2-1EL-SANIDIST-BORO      A    1
  3 GEO-WA2-1EL-SANIDIST-NUM       A    2
  2 GEO-WA2-1EL-SANITATION-SUBSEC  A    2
  2 GEO-WA2-1EL-FIRESEC            A    2
  2 GEO-WA2-1EL-FIREBAT            A    2
```

GEOLW2L COPY File (continued)

```
  2 GEO-WA2-1EL-FIRECO               A    4
R 2 GEO-WA2-1EL-FIRECO
  3 GEO-WA2-1EL-FIRECO-TYPE          A    1
  3 GEO-WA2-1EL-FIRECO-NUM           A    3
  2 GEO-WA2-1EL-SPECIAL-ADDR-FLAG    A    1
  2 GEO-WA2-1EL-MARBLE-RIKERS-FLAG   A    1
  2 GEO-WA2-1EL-SPLIT-SCHOOL-FLAG    A    1
  2 GEO-WA2-1EL-PREFERRED-LGC        A    2
  2 GEO-WA2-1EL-LIONFACECODE         A    4
  2 GEO-WA2-1EL-LIONSEQ              A    5
  2 GEO-WA2-1EL-1990-CENSUSTRACT     A    6
  2 FILLER-WA2L-480B                 A    4
  2 GEO-WA2-1EL-DYN-BLOCK            A    3
  2 GEO-WA2-1EL-XCOORD               A    7
  2 GEO-WA2-1EL-YCOORD               A    7
  2 GEO-WA2-1EL-SEGMENTLENGTH        A    5
  2 GEO-WA2-1EL-SANI-REG             A    5
  2 GEO-WA2-1EL-SEGMENT-ID           A    7
  2 GEO-WA2-1EL-TRUE-B7SC            A    8
  2 FILLER-WA2-480                   A   85
* * END OF FUNCTION 1E LONG WORKAREA LAYOUT ******
* - ------------------------------- - ---- --------------------------------
* * BEGINNING OF FUNCTION 3 LONG WORKAREA LAYOUT ******
R 1 GEOLW2L
  2 GEO-WA2-3L-ACCESS-KEY            A   21
  2 GEO-WA2-3L-DUP-KEY-FLAG          A    1 /* NOT IMPLEMENTED
  2 GEO-WA2-3L-CURVE-FLAG            A    1
  2 GEO-WA2-3L-LOCATION-STATUS       A    1
  2 GEO-WA2-3L-COUNTY-BOUNDARY       A    1
  2 FILLER-WA2-340                   A    4
  2 GEO-WA2-3L-PREFERRED-LGC1        A    2
  2 GEO-WA2-3L-PREFERRED-LGC2        A    2
  2 GEO-WA2-3L-PREFERRED-LGC3        A    2
  2 GEO-WA2-3L-NUM-X-ST-LOW-END      N    1
  2 GEO-WA2-3L-LOW-PBSC              P    6 (1:5)
  2 GEO-WA2-3L-NUM-X-ST-HI-END       N    1
  2 GEO-WA2-3L-HI-PBSC               P    6 (1:5)
  2 GEO-WA2-3L-SLA                   A    1
  2 GEO-WA2-3L-REVERSALFLAG          A    1
  2 GEO-WA2-3L-LEFT-COMDIST          A    3
R 2 GEO-WA2-3L-LEFT-COMDIST
  3 GEO-WA2-3L-LEFT-COMDIST-BORO     A    1
  3 GEO-WA2-3L-LEFT-COMDIST-NUM      A    2
  2 GEO-WA2-3L-RIGHT-COMDIST         A    3
R 2 GEO-WA2-3L-RIGHT-COMDIST
  3 GEO-WA2-3L-RIGHT-COMDIST-BORO    A    1
  3 GEO-WA2-3L-RIGHT-COMDIST-NUM     A    2
  2 GEO-WA2-3L-LEFT-ZIP              A    5
  2 GEO-WA2-3L-RIGHT-ZIP             A    5
  2 FILLER-WA2-340B                  A   18
  2 GEO-WA2-3L-LEFT-HEALTHAREA       A    4
  2 GEO-WA2-3L-RIGHT-HEALTHAREA      A    4
  2 GEO-WA2-3L-LEFT-INSTRUCT-REG     A    2
  2 GEO-WA2-3L-RIGHT-INSTRUCT-REG    A    2
  2 GEO-WA2-3L-LEFT-LOW-HOUSENUM     A    7
  2 GEO-WA2-3L-LEFT-HI-HOUSENUM      A    7
  2 GEO-WA2-3L-RIGHT-LOW-HOUSENUM    A    7
  2 GEO-WA2-3L-RIGHT-HI-HOUSENUM     A    7
  2 GEO-WA2-3L-CONT-PARITY-IND       A    1
  2 GEO-WA2-3L-LIONFACECODE          A    4
  2 GEO-WA2-3L-LIONSEQ               A    5
  2 GEO-WA2-3L-GENRECFLAG            A    1
  2 GEO-WA2-3L-SEGMENTLENGTH         P    5
  2 GEO-WA2-3L-SEGMENTSLOPE          A    3
  2 GEO-WA2-3L-SEGMENTORIENT         A    1
  2 FILLER-WA2-3L-355                A    4
```

GEOLW2L COPY File (continued)

```
  2 GEO-WA2-3L-RESDCP                A    2 /* RESERVED FOR DCP/GSS USE
  2 GEO-WA2-3L-DOG-LEG               A    1
  2 GEO-WA2-3L-FEATURE-TYPE          A    1
  2 GEO-WA2-3L-LEFT-POLICEDIST       A    4
R 2 GEO-WA2-3L-LEFT-POLICEDIST
  3 GEO-WA2-3L-L-POL-PATR-BORO-CMD   A    1
  3 GEO-WA2-3L-LEFT-POL-PRECINCT     A    3
  2 GEO-WA2-3L-RIGHT-POLICEDIST      A    4
R 2 GEO-WA2-3L-RIGHT-POLICEDIST
  3 GEO-WA2-3L-R-POL-PATR-BORO-CMD   A    1
  3 GEO-WA2-3L-RIGHT-POL-PRECINCT    A    3
  2 GEO-WA2-3L-LEFT-SCHOOLDIST       A    2
  2 GEO-WA2-3L-RIGHT-SCHOOLDIST      A    2
  2 GEO-WA2-3L-MARBLE-RIKERS-FLAG    A    1
  2 GEO-WA2-3L-SEGMENT-ID            A    7
  2 FILLER-WA2-3L-370                A    1
* * ******************************** * **** ********************************
*   THE   PORTION   OF   THIS   WORK   AREA ABOVE THIS POINT IS
*   IDENTICAL  TO  THE STANDARD WORK   AREA 2 FOR FUNCTION 3.
*   THE  PORTION  BELOW  THIS  POINT     IS PRESENT ONLY FOR THE
*   LONG WORK AREA 2 OPTION.
* * ******************************** * **** ********************************
  2 GEO-WA2-3L-L-1990-CENSUSTRACT    A    6
  2 FILLER-WA2-370B                  A    4
  2 GEO-WA2-3L-LEFT-DYN-BLK          A    3
  2 GEO-WA2-3L-R-1990-CENSUSTRACT    A    6
  2 GEO-WA2-370C                     A    4
  2 GEO-WA2-3L-RIGHT-DYN-BLK         A    3
  2 GEO-WA2-3L-LEFT-FIRESEC          A    2
  2 GEO-WA2-3L-LEFT-FIREBAT          A    2
  2 GEO-WA2-3L-LEFT-FIRECO           A    4
R 2 GEO-WA2-3L-LEFT-FIRECO
  3 GEO-WA2-3L-LEFT-FIRECO-TYPE      A    1
  3 GEO-WA2-3L-LEFT-FIRECO-NUM       A    3
  2 GEO-WA2-3L-RIGHT-FIRESEC         A    2
  2 GEO-WA2-3L-RIGHT-FIREBAT         A    2
  2 GEO-WA2-3L-RIGHT-FIRECO          A    4
R 2 GEO-WA2-3L-RIGHT-FIRECO
  3 GEO-WA2-3L-RIGHT-FIRECO-TYPE     A    1
  3 GEO-WA2-3L-RIGHT-FIRECO-NUM      A    3
  2 GEO-WA2-3L-L-2000-CENSUSTRACT    A    6
  2 GEO-WA2-3L-L-2000-CENSUSBLOCK    A    4
  2 FILLER-WA2-380B-RES-DCP          A    1
  2 GEO-WA2-3L-R-2000-CENSUSTRACT    A    6
  2 GEO-WA2-3L-R-2000-CENSUSBLOCK    A    4
  2 FILLER-WA2-W380C-RES-DCP         A    1
  2 FILLER-WA2-380                   A   36
* * END OF FUNCTION 3 LONG WORKAREA LAYOUT ******
* - -------------------------------- - ---- --------------------------------
```

GEOLW21A COPY File

```
  1 GEOLW21A                               /*FCT 1A,BL USE SAME WA2 LAYOUT
* * THE FIELD W2NAT1A IS USED AS A PARAMETER TO CALL GEOSUPPORT
  2 W2NAT1A                         A   21
R 2 W2NAT1A
  3 GEO-WA2-1A-ACCESS-KEY           A   21
  2 GEO-WA2-1A-CONT-PARITY          A    1
  2 GEO-WA2-1A-LOW-HOUSENUM         A    6
  2 GEO-WA2-1A-ALTKEY-1             A   10
R 2 GEO-WA2-1A-ALTKEY-1
  3 GEO-WA2-1A-ALTKEY-1-BORO        A    1
  3 GEO-WA2-1A-ALTKEY-1-TAXBLOCK    A    5
  3 GEO-WA2-1A-ALTKEY-1-TAXLOT      A    4
  2 FILLER-WA2-1A-230               A    1
  2 GEO-WA2-1A-SCC                  A    1
  2 FILLER-WA2-1A-240               A    1
  2 GEO-WA2-1A-GENERAL-LOT-INFO
  3 GEO-WA2-1A-RPAD-BLDG-CLASS      A    2
  3 GEO-WA2-1A-CORNER-CODE          A    2
  3 GEO-WA2-1A-NUM-OF-STRUCTURES    A    2
  3 GEO-WA2-1A-NUM-OF-BLOCKFACES    A    2
  3 GEO-WA2-1A-INTERIOR-FLAG        A    1
  3 GEO-WA2-1A-VACANT-FLAG          A    1
  3 GEO-WA2-1A-IRREG-FLAG           A    1
  2 GEO-WA2-1A-ALT-BORO-FLAG        A    1
  2 FILLER-WA2-1A-245               A    1
  2 GEO-WA2-1A-STROLL-KEY           A   13
  2 GEO-WA2-1A-OVERFLOW-FLAG        A    1
  2 FILLER-WA2-1A-251               A    1 /* USED FOR DCP
  2 GEO-WA2-1A-BIN                  A    7
  2 GEO-WA2-1A-CONDO-FLAG           A    1
  2 GEO-WA2-1A-RPAD-CONDO-NUM       A    4
  2 GEO-WA2-1A-CONDO-LOW-BBL        A   10
  2 FILLER-WA2-1A-260               A    1
  2 GEO-WA2-1A-CONDO-BILL-BBL       A   10
  2 FILLER-WA2-1A-270               A    1
  2 GEO-WA2-1A-CONDO-BILL-BBL-SCC   A    1
  2 GEO-WA2-1A-CONDO-HIGH-BBL       A   10
  2 FILLER-WA2-1A-275               A    1
  2 GEO-WA2-1A-SANBORN-BVOLPAGE     A    8
R 2 GEO-WA2-1A-SANBORN-BVOLPAGE
  3 GEO-WA2-1A-SANBORN-BORO         A    1
  3 GEO-WA2-1A-SANBORN-VOLPAGE      A    7
R 3 GEO-WA2-1A-SANBORN-VOLPAGE
  4 GEO-WA2-1A-SANBORN-VOL-NUM      A    3
  4 GEO-WA2-1A-SANBORN-VOL-PAGE     A    4
  2 GEO-WA2-1A-COMMERC-DIST         A    5
  2 GEO-WA2-1A-COOP-NUM             A    4
  2 FILLER-WA2-1A-276               A    4
  2 GEO-WA2-1A-ACTUAL-NUM-STRUCTS   A    4
  2 GEO-WA2-1A-DOF-MAP-BORO         A    1
  2 GEO-WA2-1A-DOF-MAP-SECVOL       A    4
  2 GEO-WA2-1A-DOF-MAP-PAGE         A    4
  2 GEO-WA2-1A-X-COORD              A    7
  2 GEO-WA2-1A-Y-COORD              A    7
  2 FILLER-WA2-1A-280               A   18
  2 GEO-WA2-1A-NUM-OF-ADDR-FOR-LOT  N    2
  2 GEO-WA2-1A-LIST-OF-ADDRESSES           (1:21)
  3 GEO-WA2-1A-LIST-LOW-HOUSENUM    A    6
  3 FILLER-WA2-1A-290               A    3
  3 GEO-WA2-1A-LIST-HI-HOUSENUM     A    6
  3 FILLER-WA2-1A-300               A    3
  3 GEO-WA2-1A-LIST-STREETCODE      A    8
  3 GEO-WA2-1A-LIST-BIN             A    7
  3 GEO-WA2-1A-LIST-ADDR-TYPE       A    1
  3 FILLER-WA2-1A-310               A    1
  3 GEO-WA2-1A-LIST-SOS             A    1
```

GEOLW2AL COPY File

```
  1 GEOLW2AL                                /*FCT 1A,BL USE SAME LONG WA2
* * THE FIELD W2NATAL IS USED AS A PARAMETER TO CALL GEOSUPPORT
  2 W2NATAL                         A   21
R 2 W2NATAL
  3 GEO-WA2-1AL-ACCESS-KEY          A   21
  2 GEO-WA2-1AL-CONT-PARITY         A    1
  2 GEO-WA2-1AL-LOW-HOUSENUM        A    6
  2 GEO-WA2-1AL-ALTKEY-1            A   10
R 2 GEO-WA2-1AL-ALTKEY-1
  3 GEO-WA2-1AL-ALTKEY-1-BORO       A    1
  3 GEO-WA2-1AL-ALTKEY-1-TAXBLOCK   A    5
  3 GEO-WA2-1AL-ALTKEY-1-TAXLOT     A    4
  2 FILLER-WA2-1AL-230              A    1
  2 GEO-WA2-1AL-SCC                 A    1
  2 FILLER-WA2-1AL-240              A    1
  2 GEO-WA2-1AL-GENERAL-LOT-INFO
  3 GEO-WA2-1AL-RPAD-BLDG-CLASS     A    2
  3 GEO-WA2-1AL-CORNER-CODE         A    2
  3 GEO-WA2-1AL-NUM-OF-STRUCTURES   A    2
  3 GEO-WA2-1AL-NUM-OF-BLOCKFACES   A    2
  3 GEO-WA2-1AL-INTERIOR-FLAG       A    1
  3 GEO-WA2-1AL-VACANT-FLAG         A    1
  3 GEO-WA2-1AL-IRREG-LOT-FLAG      A    1
  2 GEO-WA2-1AL-ALT-BORO-FLAG       A    1
  2 FILLER-WA2-1AL-245              A    1
  2 GEO-WA2-1AL-STROLL-KEY          A   13
  2 FILLER-WA2-1AL-250              A    1
  2 FILLER-WA2-1AL-251              A    1 /* USED FOR DCP
  2 GEO-WA2-1AL-BIN                 A    7
  2 GEO-WA2-1AL-CONDO-FLAG          A    1
  2 GEO-WA2-1AL-RPAD-CONDO-NUM      A    4
  2 GEO-WA2-1AL-CONDO-LOW-BBL       A   10
  2 FILLER-WA2-1AL-260              A    1
  2 GEO-WA2-1AL-CONDO-BILL-BBL      A   10
  2 FILLER-WA2-1AL-270              A    1
  2 GEO-WA2-1AL-CONDO-BILL-BBL-SCC  A    1
  2 GEO-WA2-1AL-CONDO-HIGH-BBL      A   10
  2 FILLER-WA2-1AL-275              A    1
  2 GEO-WA2-1AL-SANBORN-BVOLPAGE    A    8
R 2 GEO-WA2-1AL-SANBORN-BVOLPAGE
  3 GEO-WA2-1AL-SANBORN-BORO        A    1
  3 GEO-WA2-1AL-SANBORN-VOLPAGE     A    7
R 3 GEO-WA2-1AL-SANBORN-VOLPAGE
  4 GEO-WA2-1AL-SANBORN-VOL-NUM     A    3
  4 GEO-WA2-1AL-SANBORN-VOL-PAGE    A    4
  2 GEO-WA2-1AL-COMMERC-DIST        A    5
  2 GEO-WA2-1AL-COOP-NUM            A    4
  2 FILLER-WA2-1AL-276              A    4
  2 GEO-WA2-1AL-ACTUAL-NUM-STRUCTS  A    4
  2 GEO-WA2-1AL-DOF-MAP-BORO        A    1
  2 GEO-WA2-1AL-DOF-MAP-SECVOL      A    4
  2 GEO-WA2-1AL-DOF-MAP-PAGE        A    4
  2 GEO-WA2-1AL-X-COORD             A    7
  2 GEO-WA2-1AL-Y-COORD             A    7
  2 FILLER-WA2-1AL-280              A   16
  2 GEO-WA2-1AL-NUM-OF-BINS-FOR-LOT N    4
  2 GEO-WA2-1AL-LIST-OF-BINS             (1:2500)
  3 GEO-WA2-1AL-BINS                N    7
```

GEOLW23S COPY File

```
  1 GEOLW23S
* * THE FIELD W2NAT3S IS USED AS A PARAMETER TO CALL GEOSUPPORT
  2 W2NAT3S                         A   21
R 2 W2NAT3S
  3 GEO-WA2-3S-ACCESS-KEY           A   21
  2 GEO-WA2-3S-NUM-OF-INTERSECTS    N    3
  2 GEO-WA2-3S-LIST-OF-INTERSECTS           (1:350)
  3 GEO-WA2-3S-SMALLEST-PBSC        P    6
  3 GEO-WA2-3S-2ND-SMALLEST-PBSC    P    6
  3 GEO-WA2-3S-DISTANCE             P    5
  3 GEO-WA2-3S-GAP-FLAG             A    1
```

APPENDIX 6: USER FEEDBACK PROCEDURES

This appendix describes the procedures for users to provide feedback to GSS of geographic data that have either been rejected by the Geosupport System or produced unexpected results. <u>Only items that the user has reviewed and believes to be valid geographic data should be provided as feedback to GSS.</u>

Feedback from users is a crucial resource in GSS's efforts to maintain accurate and up-to-date Geosupport data files. The staff of GSS's Data Management Unit researches the feedback received from users and corrects errors and omissions in Geosupport files as appropriate. Those corrections become accessible to users when the next release of Geosupport is installed on the computer where the user's application is running.

The user should review all items and screen out those caused by obvious user data coding or data entry errors, such as an obvious street name misspelling, the specification of the intersection of two streets that are obviously parallel, etc.

The user should provide feedback on those items that the user considers to be valid data, or is uncertain about, to GSS's Data Management unit by submitting one or more completed Geosupport System User Feedback Forms. There are spaces to report up to three items on a single form. A sample form is included in this appendix and the user can replicate it as needed.

Printouts, sketch maps and/or any other material documenting the validity and location of the items should be attached to the form if possible. The user should provide any available information that would assist the GSS staff to research the issue. For example, if an address is rejected, the user should provide, if it is known, alternate addresses for the building, the names of the adjacent cross streets, the BBL (tax lot identifiers) etc.

In the case of large computer-generated reject reports, the user may submit the User Feedback Form as a transmittal form attached to the printout. In that case, it is not necessary for the user to transcribe all the reject information onto the form. If possible, the printout should display only the user input geographic data that Geosupport has rejected, not application-related data that is not passed to Geosupport. In addition, the printout should display the Geosupport Return Code, the Reason Code, and if there is space in the report, the Geosupport Message. When designing the reject report, it is advisable for the user to contact GSS's Geosupport User Liaison to ascertain how the report should be sorted. Appropriate sorting of user reject reports greatly facilitates GSS's research.

Feedback materials and inquiries about feedback procedures should be submitted to:

Geosupport System User Liaison
Department of City Planning
22 Reade Street, 4th Floor
New York, New York 10007
Phone: (212) 720-3606
FAX: (212)720-3354

Email: GSS_Feedback@planning.nyc.gov

Software issues may be directed to:

Email: GSS_Software@planning.nyc.gov

A sample User Feedback Form follows.

GEOSUPPORT SYSTEM USER FEEDBACK FORM

PAGE ________ OF ________

SUBMITTED BY: ______________________ DATE: ______________ AGENCY: ______________________ ADDRESS: ______________________ DESIGNATED CONTACT: ______________ PHONE #: ____________	NEW YORK CITY DEPARTMENT OF CITY PLANNING INFORMATION TECHNOLOGY DIVISION GEOGRAPHIC SYSTEMS SECTION DATA MANAGEMENT UNIT 22 READE STREET, 4N NEW YORK, NEW YORK 10007

ITEM # 1	FUNCTION CODE	REJECT CODE	*BORO	HOUSE NUMBER	ON STREET	1st CROSS STREET (IF KNOWN)	2nd CROSS STREET (IF KNOWN)

TAX BLOCK # (IF KNOWN)	TAX LOT # (IF KNOWN)	BUILDING ID NR. (BIN)	NR. OF BUILDINGS	SNL	COMPASS DIR.	ZIP CODE

ITEM # 2	FUNCTION CODE	REJECT CODE	*BORO	HOUSE NUMBER	ON STREET	1st CROSS STREET (IF KNOWN)	2nd CROSS STREET (IF KNOWN)

TAX BLOCK # (IF KNOWN)	TAX LOT # (IF KNOWN)	BUILDING ID NR. (BIN)	NR. OF BUILDINGS	SNL	COMPASS DIR.	ZIP CODE

ITEM # 3	FUNCTION CODE	REJECT CODE	*BORO	HOUSE NUMBER	ON STREET	1st CROSS STREET (IF KNOWN)	2nd CROSS STREET (IF KNOWN)

TAX BLOCK # (IF KNOWN)	TAX LOT # (IF KNOWN)	BUILDING ID NR. (BIN)	NR. OF BUILDINGS	SNL	COMPASS DIR.	ZIP CODE

ITEM #	USER COMMENTS	FOR DCP USE ONLY
1		
2		
3		

RETURN THIS FORM TO: GEOSUPPORT USER LIAISON PHONE # (212) 720-3606 DEPARTMENT OF CITY PLANNING FAX # (212) 720-3354 22 READE STREET (4W) NEW YORK, NEW YORK 10007 GSS_Feedback@planning.nyc.gov	PLEASE ATTACH ANY MAPS, SKETCHES OR OTHER MATERIALS WHICH MAY CLARIFY THE ABOVE ITEMS(S). *INDICATE BOROUGH AS MN, BX, BK, QN, SI	FOR DCP USE ONLY

APPENDIX 7: MAINFRAME DATA CENTERS WHERE GEOSUPPORT IS INSTALLED
as of November, 2006

NAME OF AGENCY	**LOCATION**
Department of Education (DOE)	2 MetroTech Center, Brooklyn
Department of Information Technology and Telecommunications (DoITT) / Computer Service Center (CSC)	11 MetroTech Center, Brooklyn
Financial Information Services Agency (FISA)	450 West 33 Street, Manhattan
Health and Hospitals Corporation (HHC)	230 West 41 Street, Manhattan
New York City Housing Authority (NYCHA)	250 Broadway, Manhattan
New York Police Department (NYPD)	1 Police Plaza, Manhattan

APPENDIX 8: SAMPLE APPLICATION PROGRAMS AND JCL

This appendix exhibits sample batch user application programs written in COBOL, Assembler, PL/1, C and NATURAL. These programs exemplify how a user-written application program may be coded to interact with Geosupport via its Application Programming Interface (API).

For each sample program, this appendix contains a printout of the job-stream input for an MSW application, the job-stream input for a COW application, and a printout of the program execution output report. The job-stream input contains the JCL to compile, link-edit and execute the program and, imbedded in the JCL, the program source code (except for NATURAL) and a few in-stream records of sample input data. The NATURAL program source code is not imbedded in the job-stream, and is printed separately. The MSW and COW sample programs both generate the same output report.

All of the sample programs use the Geosupport COPY facility. Since this appendix displays un-compiled source code rather than compilation output listings, the source code is shown without the COPY file expansions. Therefore, the source code as shown contains references to fields in Geosupport work areas but does not contain the definitions of those fields. To see those definitions, refer to the listings of the COPY file contents in Appendix 5 (for MSW) and Appendix 14 (for COW).

There are two sample programs in each programming language, referred to as Sample Programs #1 and #2. The processing performed in Sample Program #1 is similar for all of the programming languages, as is the processing performed in Sample Program #2. Note: there is an MSW and COW version of each of the sample programs.

In brief, Sample Program #1 reads a record containing an address from the in-stream input file; calls Functions 1 and D, checking the Geosupport Return Code (GRC) generated by each call; and writes a record into a formatted output report. The report displays the input address data and, as appropriate, selected output data obtained from Geosupport and/or the GRC, Reason Code and Message.

Sample Program #2 performs similar processing, but its input file contains street intersections instead of addresses, and accordingly it calls Function 2 instead of Function 1.

Both sample programs use the Compact Names feature to direct Geosupport to return street names in a format suitable for display in the output report. In addition, Sample Program #1, but not Sample Program #2, uses the Street Name Normalization Length Limit (SNL) feature to limit the lengths of normalized street names so they will fit into that program's output report.

In both sample programs, Function D is called to generate the cross street names. Note, that Geosupport will generate all the cross street names automatically if the user specifies the Cross Street Names Flag in the initial Function 1 or Function 2 call. (See Cross Street Names Flag in Appendix 3.) The Function D call, however, is used in the sample programs to demonstrate use of Function D and a one work-area-call.

In detail, the processing performed by Sample Program #1 is as follows:

- Read a record from the in-stream input file.

- Prime Work Area 1 with the function code, the Work Area Format Indicator (required for COW), the address information (Borough Code, House Number and Street Name) from the input record, the appropriate SNL value, and the code required to request street names in the compact format.

- Call Function 1.

- Examine the Geosupport Return Code (GRC).

- If the GRC indicates a successful call or a warning, use the street codes of the cross streets retrieved in WA2 as input to a Function D call to obtain their street names for display in the output report[3]. The Function D processing is performed as follows:

  - Prime Work Area 1 with the function code value, the Work Area Format Indicator (required for COW), and the street codes of the cross streets obtained from the Function 1 call.

  - Call Function D.

    Note: If the Cross Street Names Flag is used in the original call to Function 1, all the cross street names will be returned by Function 1.

  - Examine the GRC.

  - If the GRC is zeros, include the street names obtained from Function D in the output report. Otherwise, include the GRC, the Reason Code and the warning/reject message in the output report.

- Write an output report line containing the input information and, selected output information obtained from Work Area 2 (e.g., the Zipcode, Community District, and cross streets) and/or the Geosupport Return Code, Reason Code and Message.

---

[3] Note: The sample programs have been written in a skeletal fashion to illustrate the use of the Geosupport API as clearly as possible. Thus, for example, Sample Program #1 assumes, when it gets a 'hit' for an input address, that WA2 contains at least one cross street at each end of the block face containing the address. In reality, this is not necessarily the case; in a real application, the program would check for the presence of cross street codes before calling Function D.

COBOL SAMPLE PROGRAM #1

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

COBOL SAMPLE PROGRAM #1 - Job Stream - MSW

```
//COBF1SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  COBOL SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1  ****
//***                      MSW FORMAT                              ****
//*********************************************************************
//STEP1 EXEC   IGYWCLG,PARM.COBOL=(NOWORD,OPTIMIZE)
//COBOL.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//             DD DSN=A030.GEO.COPYLIB,DISP=SHR
//COBOL.SYSIN  DD *
      *****************************************************************
      * THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING *
      * BORO, HOUSE NUMBER & STREET NAME SUPPLIED BY AN INSTREAM FILE.*
      * FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     *
      * FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  *
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      *       NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE   *
      *             ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET *
      *             NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.     *
      *****************************************************************
      *
       IDENTIFICATION DIVISION.
         PROGRAM-ID. COBS1JOB.

      ****************************************

       ENVIRONMENT DIVISION.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
           SELECT IN-FILE ASSIGN TO INFILE.
           SELECT RPT-FILE ASSIGN TO RPTFILE.

      ****************************************

       DATA DIVISION.
         FILE SECTION.

      **** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DEFINTION *****
         FD  IN-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 80 CHARACTERS
             LABEL RECORDS ARE OMITTED.

       01  INPUT-TO-GEOSUPPORT.
           05  IN-BORO-CODE        PIC X.
           05  FILLER              PIC X.
           05  IN-HOUSE-NUMBER     PIC X(12).
           05  FILLER              PIC X.
           05  IN-STREET           PIC X(32).
           05  FILLER              PIC X(33).

         FD  RPT-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 132 CHARACTERS
             LABEL RECORDS ARE OMITTED.

       01  RPT-LINE                PIC X(132).
```

COBOL SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
 WORKING-STORAGE SECTION.

*****************************************************************
*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE ***
*** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                  ***
*****************************************************************
 01  WORK1.    COPY  W1COB.
 01  WORK2.    COPY  W2COB.

**** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT    *********

 01  RPT-DATA-LINE1.
     05  OUT-BOR             PIC X.
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-HN              PIC X(12).
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-ST              PIC X(32).
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-ZIP             PIC X(5).
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-CD              PIC X(2).
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-NYPD-PCT        PIC X(3).
     05  FILLER              PIC X(6)      VALUE ' '.
     05  OUT-SCHLDIST        PIC X(2).
     05  FILLER              PIC X(58)     VALUE ' '.

 01  RPT-DATA-LINE2.
     05  FILLER              PIC X(74)     VALUE ' '.
     05  OUT-LO-X-STREET     PIC X(25).
     05  FILLER              PIC X VALUE ' '.
     05  OUT-HI-X-STREET     PIC X(25).

 01  RPT-ERR-LINE-1.
     05  ERR-BOR             PIC X.
     05  FILLER              PIC X         VALUE ' '.
     05  ERR-HN              PIC X(12).
     05  FILLER              PIC X         VALUE ' '.
     05  ERR-ST              PIC X(32).
     05  FILLER              PIC X(14)
         VALUE ' *** FUNCTION '.
     05  ERR-FUNCTION        PIC X.
     05  FILLER              PIC X(7)
         VALUE ' GRC = '.
     05  ERR-GRC             PIC X(2).
     05  FILLER              PIC X(15) VALUE ' REASON CODE = '.
     05  ERR-REASON          PIC X.
     05  FILLER              PIC X(45) VALUE ' '.

 01  RPT-ERR-LINE-2.
     05  FILLER              PIC X(48) VALUE ' '.
     05  FILLER              PIC X(4) VALUE '*** '.
     05  OUT-ERR-MSG         PIC X(80).

 01  RPT-WRN-LINE.
     05  WRN-BOR             PIC X.
     05  FILLER              PIC X         VALUE ' '.
     05  WRN-HN              PIC X(12).
```

COBOL SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
           05  FILLER              PIC X        VALUE ' '.
           05  WRN-ST              PIC X(32).
           05  FILLER              PIC X(14)
               VALUE ' *** FUNCTION '.
           05  WRN-FUNCTION        PIC X.
           05  FILLER              PIC X(16)
               VALUE ' WARNING, GRC = '.
           05  WRN-GRC             PIC X(2).
           05  FILLER              PIC X(15) VALUE ' REASON CODE = '.
           05  WRN-REASON          PIC X(1).
           05  FILLER              PIC X(36) VALUE ' '.

       01  RPT-HEADER-1.
           05  FILLER              PIC X(40)  VALUE
           'SAMPLE COBOL PROGRAM #1 EXECUTION OUTPUT'.
           05  FILLER              PIC X(72) VALUE ' '.

       01  RPT-HEADER-2.
           05  FILLER              PIC X(58)  VALUE
           '*****---------- INPUT ADDRESS ------------***** *****-----'.
           05  FILLER              PIC X(58)  VALUE
           '---------------- SELECTED OUTPUT ITEMS -------------------'.
           05  FILLER              PIC X(16)  VALUE
           '----*****       '.

       01  RPT-HEADER-3.
           05  FILLER              PIC X(58)  VALUE
           'B HOUSE NUMBER IN-STREET-NAME                     ZIP CD N'.
           05  FILLER              PIC X(58)  VALUE
           'YPD-PCT SCHLDST LOW CROSS STREET          HIGH CROSS STREE'.
           05  FILLER              PIC X(16)  VALUE
           'T               '.

       01  RPT-HEADER-4.
           05  FILLER              PIC X(58)  VALUE
           '_ ____________ ________________________________ _____ __ _'.
           05  FILLER              PIC X(58)  VALUE
           '_______ _______ _________________________ ________________'.
           05  FILLER              PIC X(16)  VALUE
           '_________       '.

       01  FLAGS.
           05 DATA-FLAG    PIC XXX   VALUE 'YES'.
             88 MORE-DATA            VALUE 'YES'.
             88 NO-DATA              VALUE 'NO '.

      ****************************************

       PROCEDURE DIVISION.

           OPEN INPUT IN-FILE, OUTPUT RPT-FILE.
           WRITE RPT-LINE FROM RPT-HEADER-1 AFTER ADVANCING 1 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-2 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-3 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-4 AFTER ADVANCING 0 LINES.
           READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.
           PERFORM PROCESS THRU PROCESS-EX
               UNTIL NO-DATA.
```

COBOL SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
           CLOSE IN-FILE, RPT-FILE.
           MOVE 0 TO RETURN-CODE
           STOP RUN.

       PROCESS.
      *****************************************************************
      * TO MAKE A FUNCTION 1 CALL:                                    *
      *   (1) INITIALIZE WORKAREA 1 TO SPACES                         *
      *   (2) SET WA1'S FUNCTION CODE FIELD TO 1                      *
      *   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD      *
      *   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER *
      *       FIELD                                                   *
      *   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  *
      *   (6) CALL GBI WITH 2 WORKAREAS                               *
      *   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS               *
      *****************************************************************
      *****************************************************************
      * AS OF GEOSUPPORT 10.1,                                        *
      * TO RECEIVE ROADBED-SPECIFIC INFORMATION,                      *
      * SET THE ROADBED REQUEST SWITCH TO 'R', AS FOLLOWS:            *
      * MOVE 'R' TO GEO-WA1-IN-ROADBED-REQ-SWITCH.                    *
      *****************************************************************
           MOVE SPACES TO WORK1.
           MOVE '1 ' TO GEO-WA1-IN-FUNCTION-CODE.
           MOVE IN-BORO-CODE TO GEO-WA1-IN-BORO OUT-BOR ERR-BOR WRN-BOR.
           MOVE IN-HOUSE-NUMBER TO GEO-WA1-IN-HOUSENUM OUT-HN ERR-HN
                                                 WRN-HN.
           MOVE IN-STREET TO GEO-WA1-IN-STREET-1 OUT-ST ERR-ST WRN-ST.
           CALL 'GBI' USING WORK1 WORK2.

           IF GEO-WA1-OUT-RETURN-CODE NOT = 00
              MOVE '1' TO ERR-FUNCTION WRN-FUNCTION
              PERFORM PRINT-ERROR-LINE THRU P-E-EX.

           IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
             (GEO-WA1-OUT-RETURN-CODE = 01)
              PERFORM SUCCESSFUL-FUNC1 THRU S-F1-EX.

           READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.

       PROCESS-EX.
           EXIT.

       SUCCESSFUL-FUNC1.

      *********************************************************
      ***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********
      ***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS         *****
      *********************************************************
           MOVE GEO-WA2-FN1-ZIP            TO OUT-ZIP.
           MOVE GEO-WA2-FN1-COMDIST-NUMBER TO OUT-CD.
           MOVE GEO-WA2-FN1-POL-PRECINCT   TO OUT-NYPD-PCT.
           MOVE GEO-WA2-FN1-SCHOOLDIST     TO OUT-SCHLDIST.
           IF GEO-WA1-OUT-RETURN-CODE = 00
             WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 2 LINES
           ELSE
             MOVE SPACES TO OUT-BOR OUT-HN OUT-ST
             WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 1 LINES.
```

```
*************************************************************
* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND  *
* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE    *
* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS     *
* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D: *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                    *
*   (2) SET WA1'S FUNCTION CODE FIELD TO D                 *
*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED      *
*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE     *
*              HAS SPACE FOR ONLY 25 CHARACTERS)           *
*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN      *
*       STREET NAMES FORMATTED FOR DISPLAY                 *
*   (5) MOVE WA2'S LOW PBSC FIELD TO WA1'S INPUT STREET    *
*       CODE 1 FIELD                                       *
*   (6) MOVE WA2'S HIGH PBSC FIELD TO WA1'S INPUT STREET   *
*       CODE 2 FIELD                                       *
*   (7) CALL GBI WITH 1 WORKAREA                           *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS          *
*************************************************************
     MOVE SPACES TO WORK1.
     MOVE 'D ' TO GEO-WA1-IN-FUNCTION-CODE.
     MOVE 'C' TO GEO-WA1-IN-COMPACT-NAME-FLAG.
     MOVE '25' TO GEO-WA1-IN-SNL.
     MOVE GEO-WA2-FN1-LOW-PBSC (1) TO GEO-WA1-IN-STREETCODE-1.
     MOVE GEO-WA2-FN1-HI-PBSC  (1) TO GEO-WA1-IN-STREETCODE-2.
     CALL 'GBI' USING WORK1.

     IF GEO-WA1-OUT-RETURN-CODE NOT = 00
        MOVE 'D' TO ERR-FUNCTION WRN-FUNCTION
        PERFORM PRINT-ERROR-LINE THRU P-E-EX.

     IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
        (GEO-WA1-OUT-RETURN-CODE = 01)
        PERFORM SUCCESSFUL-FUNCD THRU S-FD-EX.

 S-F1-EX.
     EXIT.

 SUCCESSFUL-FUNCD.

     MOVE GEO-WA1-OUT-STREET-1 TO OUT-LO-X-STREET
     MOVE GEO-WA1-OUT-STREET-2 TO OUT-HI-X-STREET
     WRITE RPT-LINE FROM RPT-DATA-LINE2 AFTER ADVANCING 0 LINES.

 S-FD-EX.
     EXIT.


 PRINT-ERROR-LINE.
     MOVE GEO-WA1-OUT-RETURN-CODE TO ERR-GRC WRN-GRC.
     MOVE GEO-WA1-OUT-REASON-CODE TO ERR-REASON WRN-REASON.
     MOVE GEO-WA1-OUT-ERROR-MESSAGE TO OUT-ERR-MSG.

     IF GEO-WA1-OUT-RETURN-CODE = 01
**** INSERT YOUR OWN WARNING ROUTINE HERE  ****
       WRITE RPT-LINE FROM RPT-WRN-LINE AFTER ADVANCING 2 LINES
     ELSE
**** INSERT YOUR OWN WARNING ROUTINE HERE  ****
```

COBOL SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
             WRITE RPT-LINE FROM RPT-ERR-LINE-1 AFTER ADVANCING 2 LINES.

           WRITE RPT-LINE FROM RPT-ERR-LINE-2 AFTER ADVANCING 1 LINES.

       P-E-EX.
           EXIT.
/*
//LKED.SYSIN DD *
  INCLUDE INCLIB(GBI)
//LKED.INCLIB   DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//****************************************************************//
//*                                                              *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                             *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP    *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:       *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                          *//
//*       A030.GEO.SUPPORT.LOADLIB                               *//
//*                                                              *//
//****************************************************************//
//GO.STEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//****************************************************************//
//*                                                              *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                             *//
//*  DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD, ETC) *//
//*  ARE NO LONGER NEEDED AND ARE IGNORED.  GEOSUPPORT IS TAILORED *//
//*  TO USE STANDARD GEOSUPPORT DATA SET NAMES.                  *//
//*  TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.     *//
//*                                                              *//
//****************************************************************//
//*
//GO.SYSUDUMP DD SYSOUT=A,OUTLIM=3000
//GO.SYSOUT   DD SYSOUT=A
//GO.RPTFILE  DD SYSOUT=A
//GO.INFILE  DD *
1 22          READE ST
1 500         DUANE ST
1 2-4         BROADWAY
4 165-100     BAISLEY BLVD
4 165-1000    BAISLEY BLVD
/*
//
```

COBOL SAMPLE PROGRAM #1 - Job Stream - COW

```
//COBC1SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  COBOL SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1  ****
//***                    COW FORMAT                               ****
//*********************************************************************
//STEP1 EXEC   IGYWCLG,PARM.COBOL=(NOWORD,OPTIMIZE)
//COBOL.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//             DD DSN=A030.GEO.COPYLIB,DISP=SHR
//COBOL.SYSIN  DD *
      *****************************************************************
      * THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING *
      * BORO, HOUSE NUMBER & STREET NAME SUPPLIED BY AN INSTREAM FILE.*
      * FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     *
      * FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  *
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      *       NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE   *
      *             ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET *
      *             NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.     *
      *****************************************************************
      *
       IDENTIFICATION DIVISION.
         PROGRAM-ID. COBS1JOB.

      ****************************************

       ENVIRONMENT DIVISION.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
           SELECT IN-FILE ASSIGN TO INFILE.
           SELECT RPT-FILE ASSIGN TO RPTFILE.

      ****************************************

       DATA DIVISION.
         FILE SECTION.

      **** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DEFINTION *****
         FD  IN-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 80 CHARACTERS
             LABEL RECORDS ARE OMITTED.

       01  INPUT-TO-GEOSUPPORT.
           05  IN-BORO-CODE        PIC X.
           05  FILLER              PIC X.
           05  IN-HOUSE-NUMBER     PIC X(12).
           05  FILLER              PIC X.
           05  IN-STREET           PIC X(32).
           05  FILLER              PIC X(33).

         FD  RPT-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 132 CHARACTERS
             LABEL RECORDS ARE OMITTED.

       01  RPT-LINE                PIC X(132).
```

COBOL SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
 WORKING-STORAGE SECTION.

*****************************************************************
*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE ***
*** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                  ***
*****************************************************************
 01  WORK1.    COPY  P1COB.
 01  WORK2.    COPY  P2COB.

**** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT    *********

 01  RPT-DATA-LINE1.
     05  OUT-BOR              PIC X.
     05  FILLER               PIC X         VALUE ' '.
     05  OUT-HN               PIC X(12).
     05  FILLER               PIC X         VALUE ' '.
     05  OUT-ST               PIC X(32).
     05  FILLER               PIC X         VALUE ' '.
     05  OUT-ZIP              PIC X(5).
     05  FILLER               PIC X         VALUE ' '.
     05  OUT-CD               PIC X(2).
     05  FILLER               PIC X         VALUE ' '.
     05  OUT-NYPD-PCT         PIC X(3).
     05  FILLER               PIC X(6)      VALUE ' '.
     05  OUT-SCHLDIST         PIC X(2).
     05  FILLER               PIC X(58)     VALUE ' '.

 01  RPT-DATA-LINE2.
     05  FILLER               PIC X(74)     VALUE ' '.
     05  OUT-LO-X-STREET      PIC X(25).
     05  FILLER               PIC X VALUE ' '.
     05  OUT-HI-X-STREET      PIC X(25).

 01  RPT-ERR-LINE-1.
     05  ERR-BOR              PIC X.
     05  FILLER               PIC X         VALUE ' '.
     05  ERR-HN               PIC X(12).
     05  FILLER               PIC X         VALUE ' '.
     05  ERR-ST               PIC X(32).
     05  FILLER               PIC X(14)
         VALUE ' *** FUNCTION '.
     05  ERR-FUNCTION         PIC X.
     05  FILLER               PIC X(7)
         VALUE ' GRC = '.
     05  ERR-GRC              PIC X(2).
     05  FILLER               PIC X(15) VALUE ' REASON CODE = '.
     05  ERR-REASON           PIC X.
     05  FILLER               PIC X(45) VALUE ' '.

 01  RPT-ERR-LINE-2.
     05  FILLER               PIC X(48) VALUE ' '.
     05  FILLER               PIC X(4) VALUE '*** '.
     05  OUT-ERR-MSG          PIC X(80).

 01  RPT-WRN-LINE.
     05  WRN-BOR              PIC X.
     05  FILLER               PIC X         VALUE ' '.
     05  WRN-HN               PIC X(12).
```

COBOL SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
           05  FILLER              PIC X         VALUE ' '.
           05  WRN-ST              PIC X(32).
           05  FILLER              PIC X(14)
               VALUE ' *** FUNCTION '.
           05  WRN-FUNCTION        PIC X.
           05  FILLER              PIC X(16)
               VALUE ' WARNING, GRC = '.
           05  WRN-GRC             PIC X(2).
           05  FILLER              PIC X(15) VALUE ' REASON CODE = '.
           05  WRN-REASON          PIC X(1).
           05  FILLER              PIC X(36) VALUE ' '.

       01  RPT-HEADER-1.
           05  FILLER              PIC X(40)  VALUE
           'SAMPLE COBOL PROGRAM #1 EXECUTION OUTPUT'.
           05  FILLER              PIC X(72) VALUE ' '.

       01  RPT-HEADER-2.
           05  FILLER              PIC X(58)  VALUE
           '*****---------- INPUT ADDRESS ------------***** *****-----'.
           05  FILLER              PIC X(58)  VALUE
           '---------------- SELECTED OUTPUT ITEMS -------------------'.
           05  FILLER              PIC X(16)  VALUE
           '----*****       '.

       01  RPT-HEADER-3.
           05  FILLER              PIC X(58)  VALUE
           'B HOUSE NUMBER IN-STREET-NAME                     ZIP CD N'.
           05  FILLER              PIC X(58)  VALUE
           'YPD-PCT SCHLDST LOW CROSS STREET          HIGH CROSS STREE'.
           05  FILLER              PIC X(16)  VALUE
           'T               '.

       01  RPT-HEADER-4.
           05  FILLER              PIC X(58)  VALUE
           '_ ____________ ________________________________ _____ __ _'.
           05  FILLER              PIC X(58)  VALUE
           '_______ _______ _________________________ ________________'.
           05  FILLER              PIC X(16)  VALUE
           '_________       '.

       01  FLAGS.
           05 DATA-FLAG    PIC XXX   VALUE 'YES'.
             88 MORE-DATA            VALUE 'YES'.
             88 NO-DATA              VALUE 'NO '.

      ****************************************

       PROCEDURE DIVISION.

           OPEN INPUT IN-FILE, OUTPUT RPT-FILE.
           WRITE RPT-LINE FROM RPT-HEADER-1 AFTER ADVANCING 1 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-2 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-3 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-4 AFTER ADVANCING 0 LINES.
           READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.
           PERFORM PROCESS THRU PROCESS-EX
               UNTIL NO-DATA.
```

```
           CLOSE IN-FILE, RPT-FILE.
           MOVE 0 TO RETURN-CODE
           STOP RUN.

       PROCESS.
      *****************************************************************
      * TO MAKE A FUNCTION 1 CALL:                                    *
      *   (1) INITIALIZE WORKAREA 1 TO SPACES                         *
      *   (2) SET WA1'S FUNCTION CODE FIELD TO 1                      *
      *   (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)   *
      *       TO USE CHARACTER-ONLY WORK AREAS (COWS)                 *
      *   (4) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD      *
      *   (5) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER *
      *       FIELD                                                   *
      *   (6) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  *
      *   (7) CALL GBI WITH 2 WORKAREAS                               *
      *   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS               *
      *****************************************************************
      *****************************************************************
      * AS OF GEOSUPPORT 10.1,                                        *
      * TO RECEIVE ROADBED-SPECIFIC INFORMATION,                      *
      * SET THE ROADBED REQUEST SWITCH TO 'R', AS FOLLOWS:            *
      * MOVE 'R' TO PIWA1-IN-ROADBED-REQ-SWITCH.                      *
      *****************************************************************
           MOVE SPACES TO WORK1.
           MOVE '1 ' TO PIWA1-IN-FUNC-CODE.
           MOVE 'C' TO GEO-WA1-IN-NON-IBM-MAIN-FRAME.
           MOVE IN-BORO-CODE TO GEO-WA1-IN-BORO OUT-BOR ERR-BOR WRN-BOR.
           MOVE IN-HOUSE-NUMBER TO PIWA1-IN-HOUSENUM-DISPLAY OUT-HN
                                                ERR-HN WRN-HN.
           MOVE IN-STREET TO GEO-WA1-IN-STREET-1 OUT-ST ERR-ST WRN-ST.
           CALL 'GBI' USING WORK1 WORK2.

           IF GEO-WA1-OUT-RETURN-CODE NOT = 00
              MOVE '1' TO ERR-FUNCTION WRN-FUNCTION
              PERFORM PRINT-ERROR-LINE THRU P-E-EX.

           IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
             (GEO-WA1-OUT-RETURN-CODE = 01)
              PERFORM SUCCESSFUL-FUNC1 THRU S-F1-EX.

           READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.

       PROCESS-EX.
           EXIT.

       SUCCESSFUL-FUNC1.

      ***********************************************************
      ***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********
      ***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS         *****
      ***********************************************************
           MOVE GEO-WA2-FN1-ZIP            TO OUT-ZIP.
           MOVE GEO-WA2-FN1-COMDIST-NUMBER TO OUT-CD.
           MOVE GEO-WA2-FN1-POL-PRECINCT   TO OUT-NYPD-PCT.
           MOVE GEO-WA2-FN1-SCHOOLDIST     TO OUT-SCHLDIST.
           IF GEO-WA1-OUT-RETURN-CODE = 00
             WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 2 LINES
```

```
     ELSE
       MOVE SPACES TO OUT-BOR OUT-HN OUT-ST
       WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 1 LINES.

****************************************************************
* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND      *
* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE        *
* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS         *
* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D:     *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                        *
*   (2) SET WA1'S FUNCTION CODE FIELD TO D                     *
*   (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)  *
*       TO USE CHARACTER-ONLY WORK AREAS (COWS)                *
*   (4) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED          *
*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE         *
*              HAS SPACE FOR ONLY 25 CHARACTERS)               *
*   (5) USE THE COMPACT STREET NAMES OPTION TO OBTAIN          *
*       STREET NAMES FORMATTED FOR DISPLAY                     *
*   (6) MOVE WA2'S LOW BSC FIELD TO WA1'S INPUT STREET         *
*       CODE 1 FIELD                                           *
*   (7) MOVE WA2'S HIGH BSC FIELD TO WA1'S INPUT STREET        *
*       CODE 2 FIELD                                           *
*   (8) CALL GBI WITH 1 WORKAREA                               *
*   (9) CHECK RETURN CODES FOR ERRORS OR WARNINGS              *
****************************************************************
     MOVE SPACES TO WORK1.
     MOVE 'D ' TO PIWA1-IN-FUNC-CODE.
     MOVE 'C' TO GEO-WA1-IN-NON-IBM-MAIN-FRAME.
     MOVE 'C' TO GEO-WA1-IN-COMPACT-NAME-FLAG.
     MOVE '25' TO GEO-WA1-IN-SNL.
     MOVE PIWA2-FN1-LOW-B5SC (1) TO GEO-WA1-IN-10SC-1.
     MOVE PIWA2-FN1-HI-B5SC  (1) TO GEO-WA1-IN-10SC-2.
     CALL 'GBI' USING WORK1.

     IF GEO-WA1-OUT-RETURN-CODE NOT = 00
        MOVE 'D' TO ERR-FUNCTION WRN-FUNCTION
        PERFORM PRINT-ERROR-LINE THRU P-E-EX.

     IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
        (GEO-WA1-OUT-RETURN-CODE = 01)
        PERFORM SUCCESSFUL-FUNCD THRU S-FD-EX.

 S-F1-EX.
     EXIT.

 SUCCESSFUL-FUNCD.

     MOVE GEO-WA1-OUT-STREET-1 TO OUT-LO-X-STREET
     MOVE GEO-WA1-OUT-STREET-2 TO OUT-HI-X-STREET
     WRITE RPT-LINE FROM RPT-DATA-LINE2 AFTER ADVANCING 0 LINES.

 S-FD-EX.
     EXIT.


 PRINT-ERROR-LINE.
     MOVE GEO-WA1-OUT-RETURN-CODE TO ERR-GRC WRN-GRC.
     MOVE GEO-WA1-OUT-REASON-CODE TO ERR-REASON WRN-REASON.
```

COBOL SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
           MOVE GEO-WA1-OUT-ERROR-MESSAGE TO OUT-ERR-MSG.

           IF GEO-WA1-OUT-RETURN-CODE = 01
      **** INSERT YOUR OWN WARNING ROUTINE HERE  ****
             WRITE RPT-LINE FROM RPT-WRN-LINE AFTER ADVANCING 2 LINES
           ELSE
      **** INSERT YOUR OWN ERROR ROUTINE HERE  ****
             WRITE RPT-LINE FROM RPT-ERR-LINE-1 AFTER ADVANCING 2 LINES.

           WRITE RPT-LINE FROM RPT-ERR-LINE-2 AFTER ADVANCING 1 LINES.

       P-E-EX.
           EXIT.
/*
//LKED.SYSIN DD *
  INCLUDE INCLIB(GBI)
//LKED.INCLIB   DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//******************************************************************//
//*                                                                *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                               *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP      *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:         *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                            *//
//*       A030.GEO.SUPPORT.LOADLIB                                 *//
//*                                                                *//
//******************************************************************//
//GO.STEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//******************************************************************//
//*                                                                *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                               *//
//*  DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD, ETC) *//
//*  ARE NO LONGER NEEDED AND ARE IGNORED.  GEOSUPPORT IS TAILORED *//
//*  TO USE STANDARD GEOSUPPORT DATA SET NAMES.                    *//
//*  TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.       *//
//*                                                                *//
//******************************************************************//
//*
//GO.SYSUDUMP DD SYSOUT=A,OUTLIM=3000
//GO.SYSOUT   DD SYSOUT=A
//GO.RPTFILE  DD SYSOUT=A
//GO.INFILE  DD *
1 22            READE ST
1 500           DUANE ST
1 2-4           BROADWAY
4 165-100       BAISLEY BLVD
4 165-1000      BAISLEY BLVD
/*
//
```

COBOL SAMPLE PROGRAM #1 - Output Report

```
SAMPLE COBOL PROGRAM #1 EXECUTION OUTPUT

*****---------- INPUT ADDRESS ------------***** *****--------------------- SELECTED OUTPUT ITEMS -----------------------*****

B HOUSE NUMBER IN-STREET-NAME                    ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET          HIGH CROSS STREET
- ------------ -------------------------------- ----- -- -------- ------- ------------------------- -------------------------

1 22           READE ST                         10007 01 005      02      ELK STREET                BROADWAY

1 500          DUANE ST                         *** FUNCTION 1 GRC = 42 REASON CODE =
                                                *** ADDRESS NUMBER OUT OF RANGE

1 2-4          BROADWAY                         *** FUNCTION 1 WARNING, GRC = 01 REASON CODE = 1
                                                *** ADDR NUMBER ALTERED: RANGE ASSUMED. USING DIGITS BEFORE DASH ONLY
                                                10004 01 001      02      STONE STREET              BOWLING GREEN

4 165-100      BAISLEY BLVD                     11434 12 113      28     GUY R BREWER BOULEVARD    BEDELL STREET

4 165-1000     BAISLEY BLVD                     *** FUNCTION 1 GRC = 13 REASON CODE = 2
                                                *** ADDRESS NBR 165-1000  HAS MORE THAN 3 DIGITS AFTER THE DASH.
```

COBOL SAMPLE PROGRAM #2

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

COBOL SAMPLE PROGRAM #2 - Job Stream - MSW

```
//COBF2SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  COBOL SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2  ****
//***                    MSW FORMAT                               ****
//*********************************************************************
//STEP1 EXEC   IGYWCLG,PARM.COBOL=(NOWORD,OPTIMIZE)
//COBOL.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//             DD DSN=A030.GEO.COPYLIB,DISP=SHR
//COBOL.SYSIN  DD *
      ****************************************************************
      * THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING *
      * TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.  *
      * FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION.*
      * FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  *
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      *       NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE   *
      *             ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET *
      *             NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 2.     *
      ****************************************************************
      *
       IDENTIFICATION DIVISION.
         PROGRAM-ID. COBS1JOB.

      ****************************************

       ENVIRONMENT DIVISION.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
           SELECT IN-FILE ASSIGN TO INFILE.
           SELECT RPT-FILE ASSIGN TO RPTFILE.

      ****************************************

       DATA DIVISION.
         FILE SECTION.

      **** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DEFINTION *****
         FD  IN-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 80 CHARACTERS
             LABEL RECORDS ARE OMITTED.

       01  INPUT-TO-GEOSUPPORT.
           05  IN-BOR1             PIC X.
           05  FILLER              PIC X.
           05  IN-STREET1          PIC X(32).
           05  FILLER              PIC X.
           05  IN-BOR2             PIC X.
           05  FILLER              PIC X.
           05  IN-STREET2          PIC X(32).
           05  FILLER              PIC X(11).

         FD  RPT-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 132 CHARACTERS
             LABEL RECORDS ARE OMITTED.
```

COBOL SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
 01  RPT-LINE               PIC X(132).

 WORKING-STORAGE SECTION.

 77  I                      PIC 9 VALUE 0.

*****************************************************************
*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE ***
*** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                  ***
*****************************************************************
 01  WORK1.    COPY  W1COB.
 01  WORK2.    COPY  W2COB.

**** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT    *********

 01  RPT-DATA-LINE1.
     05  OUT-BOR1           PIC X.
     05  FILLER             PIC X         VALUE ' '.
     05  OUT-ST1            PIC X(32).
     05  FILLER             PIC X         VALUE ' '.
     05  OUT-BOR2           PIC X.
     05  FILLER             PIC X         VALUE ' '.
     05  OUT-ST2            PIC X(32).
     05  OUT-DETAIL.
       10  FILLER           PIC X         VALUE ' '.
       10  OUT-ZIP          PIC X(5).
       10  FILLER           PIC X         VALUE ' '.
       10  OUT-CD           PIC X(2).
       10  FILLER           PIC X         VALUE ' '.
       10  OUT-NYPD-PCT     PIC X(3).
       10  FILLER           PIC X(6)      VALUE ' '.
       10  OUT-SCHLDIST     PIC X(2).
       10  FILLER           PIC X(42)     VALUE ' '.

 01  RPT-DATA-LINE2.
     05  FILLER             PIC X(96)     VALUE ' '.
     05  OUT-ST             PIC X(32).
     05  FILLER             PIC X(4) VALUE ' '.

 01  RPT-ERR-LINE.
     05  FILLER             PIC X(15) VALUE '***** FUNCTION '.
     05  ERR-FUNCTION       PIC X.
     05  FILLER             PIC X(7)  VALUE ' GRC = '.
     05  ERR-GRC            PIC X(2).
     05  FILLER             PIC X(15) VALUE ' REASON CODE = '.
     05  ERR-REASON         PIC X.
     05  FILLER             PIC X(2)  VALUE '. '.
     05  OUT-ERR-MSG        PIC X(80).
     05  FILLER             PIC X(9) VALUE ' '.

 01  RPT-WRN-LINE.
     05  FILLER             PIC X(15) VALUE '***** FUNCTION '.
     05  WRN-FUNCTION       PIC X.
     05  FILLER             PIC X(15) VALUE ' WARNING GRC = '.
     05  WRN-GRC            PIC X(2).
     05  FILLER             PIC X(15) VALUE ' REASON CODE = '.
     05  WRN-REASON         PIC X.
     05  FILLER             PIC X(2)  VALUE '. '.
```

COBOL SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
           05  OUT-WRN-MSG         PIC X(80).
           05  FILLER              PIC X    VALUE ' '.

       01  RPT-HEADER-1.
           05  FILLER              PIC X(40)  VALUE
           'SAMPLE COBOL PROGRAM #2 EXECUTION OUTPUT'.
           05  FILLER              PIC X(72) VALUE ' '.

       01  RPT-HEADER-2.
           05  FILLER              PIC X(58)  VALUE
           '*****----------------- INPUT INTERSECTION ----------------'.
           05  FILLER              PIC X(58)  VALUE
           '------***** *****------------ SELECTED OUTPUT ITEMS ------'.
           05  FILLER              PIC X(16)  VALUE
           '-------*****'.

       01  RPT-HEADER-3.
           05  FILLER              PIC X(58)  VALUE
           'B IN-STREET-NAME-1                 B IN-STREET-NAME-2     '.
           05  FILLER              PIC X(58)  VALUE
           '              ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET '.
           05  FILLER              PIC X(16)  VALUE
           'NAMES           '.

       01  RPT-HEADER-4.
           05  FILLER              PIC X(58)  VALUE
           '_ ________________________________ _ _____________________'.
           05  FILLER              PIC X(58)  VALUE
           '___________ _____ __ ________ _______ ____________________'.
           05  FILLER              PIC X(16)  VALUE
           '____________    '.

       01  FLAGS.
           05 DATA-FLAG    PIC XXX   VALUE 'YES'.
             88 MORE-DATA            VALUE 'YES'.
             88 NO-DATA              VALUE 'NO '.

      ****************************************

       PROCEDURE DIVISION.

           OPEN INPUT IN-FILE, OUTPUT RPT-FILE.
           WRITE RPT-LINE FROM RPT-HEADER-1 AFTER ADVANCING 1 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-2 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-3 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-4 AFTER ADVANCING 0 LINES.
           READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.
           PERFORM PROCESS THRU PROCESS-EX
               UNTIL NO-DATA.
           CLOSE IN-FILE, RPT-FILE.
           MOVE 0 TO RETURN-CODE
           STOP RUN.

       PROCESS.
      *****************************************************************
      * TO MAKE A FUNCTION 2 CALL:                                    *
      *   (1) INITIALIZE WORKAREA 1 TO SPACES                         *
      *   (2) SET WA1'S FUNCTION-CODE TO 2                            *
```

COBOL SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   *
*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     *
*       FIELD                                                    *
*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD *
*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   *
*       FIELD                                                    *
*   (7) CALL GBI WITH 2 WORKAREAS                                *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS                *
******************************************************************
     MOVE SPACES TO WORK1.
     MOVE '2 ' TO GEO-WA1-IN-FUNCTION-CODE.
     MOVE IN-BOR1 TO GEO-WA1-IN-BORO   OUT-BOR1.
     MOVE IN-BOR2 TO GEO-WA1-IN-BORO-2 OUT-BOR2.
     MOVE IN-STREET1 TO GEO-WA1-IN-STREET-1 OUT-ST1.
     MOVE IN-STREET2 TO GEO-WA1-IN-STREET-2 OUT-ST2.
     CALL 'GBI' USING WORK1 WORK2.

     IF GEO-WA1-OUT-RETURN-CODE NOT = 00
        MOVE '2' TO ERR-FUNCTION WRN-FUNCTION
        PERFORM PRINT-ERROR-LINE THRU P-E-EX.

     IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
       (GEO-WA1-OUT-RETURN-CODE = 01)
        PERFORM SUCCESSFUL-FUNC2 THRU S-F2-EX
     ELSE
       MOVE SPACES TO OUT-DETAIL
       WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 1 LINES.

     READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.

 PROCESS-EX.
     EXIT.

 SUCCESSFUL-FUNC2.

***********************************************************
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********
***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS          *****
***********************************************************
     MOVE GEO-WA2-FN2-ZIP            TO OUT-ZIP.
     MOVE GEO-WA2-FN2-COMDIST-NUMBER TO OUT-CD.
     MOVE GEO-WA2-FN2-POL-PRECINCT   TO OUT-NYPD-PCT.
     MOVE GEO-WA2-FN2-SCHOOLDIST     TO OUT-SCHLDIST.
* PROCESS CROSS STREET** CHECK FOR AT LEAST 1.
     WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 2 LINES.

     PERFORM CALL-D THRU CALL-D-EX
        VARYING I FROM 1 BY 1 UNTIL
            (I > GEO-WA2-FN2-NUM-OF-INTERSECTS).

 S-F2-EX.
     EXIT.

 CALL-D.

*******************************************************
* TO GET STREET NAMES FOR INTERSECTING STREET CODES   *
* MAKE A FUNCTION D CALL:                             *
```

COBOL SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
      *   (1) INITIALIZE WORKAREA 1 TO SPACES              *
      *   (2) SET THE WA1'S FUNCTION CODE FIELD TO D       *
      *   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN *
      *       STREET NAMES FORMATTED FOR DISPLAY           *
      *   (4) MOVE THE PACKED BORO AND STREET CODE TO      *
      *       WA1'S INPUT STREET CODE 1 FIELD              *
      *   (5) CALL GBI WITH 1 WORKAREA                     *
      *   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS    *
      *******************************************************
           MOVE SPACES TO WORK1.
           MOVE 'D ' TO GEO-WA1-IN-FUNCTION-CODE.
           MOVE 'C' TO GEO-WA1-IN-COMPACT-NAME-FLAG.
           MOVE '25' TO GEO-WA1-IN-SNL.
           MOVE GEO-WA2-FN2-INTERSECT-PBSC (I)
               TO GEO-WA1-IN-STREETCODE-1
           CALL 'GBI' USING WORK1.

           IF GEO-WA1-OUT-RETURN-CODE NOT = 00
              MOVE 'D' TO ERR-FUNCTION WRN-FUNCTION
              PERFORM PRINT-ERROR-LINE THRU P-E-EX.

           IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
              (GEO-WA1-OUT-RETURN-CODE = 01)
              PERFORM SUCCESSFUL-FUNCD THRU S-FD-EX.

       CALL-D-EX.
           EXIT.

       SUCCESSFUL-FUNCD.

             MOVE GEO-WA1-OUT-STREET-1 TO OUT-ST
             IF I = 1
             WRITE RPT-LINE FROM RPT-DATA-LINE2 AFTER ADVANCING 0 LINES
             ELSE
             WRITE RPT-LINE FROM RPT-DATA-LINE2 AFTER ADVANCING 1 LINES.

       S-FD-EX.
           EXIT.


       PRINT-ERROR-LINE.
           MOVE GEO-WA1-OUT-RETURN-CODE TO ERR-GRC WRN-GRC.
           MOVE GEO-WA1-OUT-REASON-CODE TO ERR-REASON WRN-REASON.
           MOVE GEO-WA1-OUT-ERROR-MESSAGE TO OUT-ERR-MSG OUT-WRN-MSG.

           IF GEO-WA1-OUT-RETURN-CODE = 01
      **** INSERT YOUR OWN WARNING ROUTINE HERE  ****
             WRITE RPT-LINE FROM RPT-WRN-LINE AFTER ADVANCING 2 LINES
           ELSE
      **** INSERT YOUR OWN WARNING ROUTINE HERE  ****
             WRITE RPT-LINE FROM RPT-ERR-LINE AFTER ADVANCING 2 LINES.

       P-E-EX.
           EXIT.
/*
//LKED.SYSIN DD *
  INCLUDE INCLIB(GBI)
//LKED.INCLIB   DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
```

COBOL SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
//*****************************************************************//
//*                                                               *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                              *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP     *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:        *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                           *//
//*       A030.GEO.SUPPORT.LOADLIB                                *//
//*                                                               *//
//*****************************************************************//
//GO.STEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//*****************************************************************//
//*                                                               *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                              *//
//*  DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD, ETC) *//
//*  ARE NO LONGER NEEDED AND ARE IGNORED.  GEOSUPPORT IS TAILORED *//
//*  TO USE STANDARD GEOSUPPORT DATA SET NAMES.                   *//
//*  TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.      *//
//*                                                               *//
//*****************************************************************//
//*
//GO.SYSUDUMP DD SYSOUT=A,OUTLIM=3000
//GO.SYSOUT   DD SYSOUT=A
//GO.RPTFILE  DD SYSOUT=A
//GO.INFILE  DD *
1 CHAMBERS ST                      1 HUDSON ST
1 SIXTH AVE                        1 W. 8 ST
1 DUANE ST                         1 READE ST
/*
//
```

COBOL SAMPLE PROGRAM #2 - Job Stream - COW

```
//COBC2SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  COBOL SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2  ****
//***                     COW FORMAT                              ****
//*********************************************************************
//STEP1 EXEC   IGYWCLG,PARM.COBOL=(NOWORD,OPTIMIZE)
//COBOL.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//             DD DSN=A030.GEO.COPYLIB,DISP=SHR
//COBOL.SYSIN  DD *
      *****************************************************************
      * THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING *
      * TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.  *
      * FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION.*
      * FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  *
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      *      NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE    *
      *            ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET *
      *            NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 2.     *
      *****************************************************************
      *
       IDENTIFICATION DIVISION.
         PROGRAM-ID. COBS1JOB.

      ****************************************

       ENVIRONMENT DIVISION.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
           SELECT IN-FILE ASSIGN TO INFILE.
           SELECT RPT-FILE ASSIGN TO RPTFILE.

      ****************************************

       DATA DIVISION.
         FILE SECTION.

      **** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DEFINTION *****
         FD  IN-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 80 CHARACTERS
             LABEL RECORDS ARE OMITTED.

       01  INPUT-TO-GEOSUPPORT.
           05  IN-BOR1              PIC X.
           05  FILLER               PIC X.
           05  IN-STREET1           PIC X(32).
           05  FILLER               PIC X.
           05  IN-BOR2              PIC X.
           05  FILLER               PIC X.
           05  IN-STREET2           PIC X(32).
           05  FILLER               PIC X(11).

         FD  RPT-FILE
             RECORDING MODE IS F
             RECORD CONTAINS 132 CHARACTERS
             LABEL RECORDS ARE OMITTED.
```

COBOL SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
 01  RPT-LINE                PIC X(132).

 WORKING-STORAGE SECTION.

 77  I                       PIC 9 VALUE 0.

*****************************************************************
*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE ***
*** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                  ***
*****************************************************************
 01  WORK1.    COPY  P1COB.
 01  WORK2.    COPY  P2COB.

**** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT    *********

 01  RPT-DATA-LINE1.
     05  OUT-BOR1            PIC X.
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-ST1             PIC X(32).
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-BOR2            PIC X.
     05  FILLER              PIC X         VALUE ' '.
     05  OUT-ST2             PIC X(32).
     05  OUT-DETAIL.
       10  FILLER            PIC X         VALUE ' '.
       10  OUT-ZIP           PIC X(5).
       10  FILLER            PIC X         VALUE ' '.
       10  OUT-CD            PIC X(2).
       10  FILLER            PIC X         VALUE ' '.
       10  OUT-NYPD-PCT      PIC X(3).
       10  FILLER            PIC X(6)      VALUE ' '.
       10  OUT-SCHLDIST      PIC X(2).
       10  FILLER            PIC X(42)     VALUE ' '.

 01  RPT-DATA-LINE2.
     05  FILLER              PIC X(96)     VALUE ' '.
     05  OUT-ST              PIC X(32).
     05  FILLER              PIC X(4) VALUE ' '.

 01  RPT-ERR-LINE.
     05  FILLER              PIC X(15) VALUE '***** FUNCTION '.
     05  ERR-FUNCTION        PIC X.
     05  FILLER              PIC X(7)  VALUE ' GRC = '.
     05  ERR-GRC             PIC X(2).
     05  FILLER              PIC X(15) VALUE ' REASON CODE = '.
     05  ERR-REASON          PIC X.
     05  FILLER              PIC X(2)  VALUE '. '.
     05  OUT-ERR-MSG         PIC X(80).
     05  FILLER              PIC X(9) VALUE ' '.

 01  RPT-WRN-LINE.
     05  FILLER              PIC X(15) VALUE '***** FUNCTION '.
     05  WRN-FUNCTION        PIC X.
     05  FILLER              PIC X(15) VALUE ' WARNING GRC = '.
     05  WRN-GRC             PIC X(2).
     05  FILLER              PIC X(15) VALUE ' REASON CODE = '.
     05  WRN-REASON          PIC X.
     05  FILLER              PIC X(2)  VALUE '. '.
```

COBOL SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
           05  OUT-WRN-MSG          PIC X(80).
           05  FILLER               PIC X    VALUE ' '.

       01  RPT-HEADER-1.
           05  FILLER               PIC X(40)  VALUE
           'SAMPLE COBOL PROGRAM #2 EXECUTION OUTPUT'.
           05  FILLER               PIC X(72) VALUE ' '.

       01  RPT-HEADER-2.
           05  FILLER               PIC X(58)  VALUE
           '*****----------------- INPUT INTERSECTION ----------------'.
           05  FILLER               PIC X(58)  VALUE
           '------***** *****------------ SELECTED OUTPUT ITEMS ------'.
           05  FILLER               PIC X(16)  VALUE
           '-------*****'.

       01  RPT-HEADER-3.
           05  FILLER               PIC X(58)  VALUE
           'B IN-STREET-NAME-1                 B IN-STREET-NAME-2     '.
           05  FILLER               PIC X(58)  VALUE
           '              ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET '.
           05  FILLER               PIC X(16)  VALUE
           'NAMES           '.

       01  RPT-HEADER-4.
           05  FILLER               PIC X(58)  VALUE
           '_ ________________________________ _ _____________________'.
           05  FILLER               PIC X(58)  VALUE
           '___________ _____ __ ________ _______ ____________________'.
           05  FILLER               PIC X(16)  VALUE
           '____________    '.

       01  FLAGS.
           05 DATA-FLAG    PIC XXX   VALUE 'YES'.
             88 MORE-DATA            VALUE 'YES'.
             88 NO-DATA              VALUE 'NO '.

      ****************************************

       PROCEDURE DIVISION.

           OPEN INPUT IN-FILE, OUTPUT RPT-FILE.
           WRITE RPT-LINE FROM RPT-HEADER-1 AFTER ADVANCING 1 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-2 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-3 AFTER ADVANCING 2 LINES.
           WRITE RPT-LINE FROM RPT-HEADER-4 AFTER ADVANCING 0 LINES.
           READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.
           PERFORM PROCESS THRU PROCESS-EX
               UNTIL NO-DATA.
           CLOSE IN-FILE, RPT-FILE.
           MOVE 0 TO RETURN-CODE
           STOP RUN.

       PROCESS.
      ******************************************************************
      * TO MAKE A FUNCTION 2 CALL:                                     *
      *   (1) INITIALIZE WORKAREA 1 TO SPACES                          *
      *   (2) SET WA1'S FUNCTION-CODE TO 2                             *
```

```
*   (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)    *
*       TO USE CHARACTER-ONLY WORK AREAS (COWS)                  *
*   (4) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   *
*   (5) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     *
*       FIELD                                                    *
*   (6) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD *
*   (7) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   *
*       FIELD                                                    *
*   (8) CALL GBI WITH 2 WORKAREAS                                *
*   (9) CHECK RETURN CODES FOR ERRORS OR WARNINGS                *
******************************************************************
     MOVE SPACES TO WORK1.
     MOVE '2 ' TO PIWA1-IN-FUNC-CODE.
     MOVE 'C' TO GEO-WA1-IN-NON-IBM-MAIN-FRAME.
     MOVE IN-BOR1 TO GEO-WA1-IN-BORO   OUT-BOR1.
     MOVE IN-BOR2 TO GEO-WA1-IN-BORO-2 OUT-BOR2.
     MOVE IN-STREET1 TO GEO-WA1-IN-STREET-1 OUT-ST1.
     MOVE IN-STREET2 TO GEO-WA1-IN-STREET-2 OUT-ST2.
     CALL 'GBI' USING WORK1 WORK2.

     IF GEO-WA1-OUT-RETURN-CODE NOT = 00
        MOVE '2' TO ERR-FUNCTION WRN-FUNCTION
        PERFORM PRINT-ERROR-LINE THRU P-E-EX.

     IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
       (GEO-WA1-OUT-RETURN-CODE = 01)
        PERFORM SUCCESSFUL-FUNC2 THRU S-F2-EX
     ELSE
       MOVE SPACES TO OUT-DETAIL
       WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 1 LINES.

     READ IN-FILE AT END MOVE 'NO ' TO DATA-FLAG.

 PROCESS-EX.
     EXIT.

 SUCCESSFUL-FUNC2.

***********************************************************
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********
***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS         *****
***********************************************************
     MOVE GEO-WA2-FN2-ZIP            TO OUT-ZIP.
     MOVE GEO-WA2-FN2-COMDIST-NUMBER TO OUT-CD.
     MOVE GEO-WA2-FN2-POL-PRECINCT   TO OUT-NYPD-PCT.
     MOVE GEO-WA2-FN2-SCHOOLDIST     TO OUT-SCHLDIST.
* PROCESS CROSS STREET** CHECK FOR AT LEAST 1.
     WRITE RPT-LINE FROM RPT-DATA-LINE1 AFTER ADVANCING 2 LINES.

     PERFORM CALL-D THRU CALL-D-EX
        VARYING I FROM 1 BY 1 UNTIL
            (I > GEO-WA2-FN2-NUM-OF-INTERSECTS).

 S-F2-EX.
     EXIT.

 CALL-D.
```

COBOL SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
*****************************************************************
* TO GET STREET NAMES FOR INTERSECTING STREET CODES             *
* MAKE A FUNCTION D CALL:                                        *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                          *
*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D                   *
*   (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)    *
*       TO USE CHARACTER-ONLY WORK AREAS (COWS)                  *
*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN            *
*       STREET NAMES FORMATTED FOR DISPLAY                       *
*   (5) MOVE THE BORO AND STREET CODE                            *
*       WA1'S INPUT STREET CODE 1 FIELD                          *
*   (6) CALL GBI WITH 1 WORKAREA                                 *
*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS                *
*****************************************************************
     MOVE SPACES TO WORK1.
     MOVE 'D ' TO PIWA1-IN-FUNC-CODE.
     MOVE 'C' TO GEO-WA1-IN-NON-IBM-MAIN-FRAME.
     MOVE 'C' TO GEO-WA1-IN-COMPACT-NAME-FLAG.
     MOVE '25' TO GEO-WA1-IN-SNL.
     MOVE PIWA2-FN2-INTERSECT-B5SC (I)
         TO GEO-WA1-IN-10SC-1
     CALL 'GBI' USING WORK1.

     IF GEO-WA1-OUT-RETURN-CODE NOT = 00
        MOVE 'D' TO ERR-FUNCTION WRN-FUNCTION
        PERFORM PRINT-ERROR-LINE THRU P-E-EX.

     IF (GEO-WA1-OUT-RETURN-CODE = 00) OR
        (GEO-WA1-OUT-RETURN-CODE = 01)
        PERFORM SUCCESSFUL-FUNCD THRU S-FD-EX.

 CALL-D-EX.
     EXIT.

 SUCCESSFUL-FUNCD.

       MOVE GEO-WA1-OUT-STREET-1 TO OUT-ST
       IF I = 1
       WRITE RPT-LINE FROM RPT-DATA-LINE2 AFTER ADVANCING 0 LINES
       ELSE
       WRITE RPT-LINE FROM RPT-DATA-LINE2 AFTER ADVANCING 1 LINES.

 S-FD-EX.
     EXIT.


 PRINT-ERROR-LINE.
     MOVE GEO-WA1-OUT-RETURN-CODE TO ERR-GRC WRN-GRC.
     MOVE GEO-WA1-OUT-REASON-CODE TO ERR-REASON WRN-REASON.
     MOVE GEO-WA1-OUT-ERROR-MESSAGE TO OUT-ERR-MSG OUT-WRN-MSG.

     IF GEO-WA1-OUT-RETURN-CODE = 01
**** INSERT YOUR OWN WARNING ROUTINE HERE  ****
       WRITE RPT-LINE FROM RPT-WRN-LINE AFTER ADVANCING 2 LINES
     ELSE
**** INSERT YOUR OWN WARNING ROUTINE HERE  ****
       WRITE RPT-LINE FROM RPT-ERR-LINE AFTER ADVANCING 2 LINES.
```

COBOL SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
       P-E-EX.
           EXIT.
/*
//LKED.SYSIN DD *
  INCLUDE INCLIB(GBI)
//LKED.INCLIB   DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************//
//*                                                               *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                              *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP     *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:        *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                           *//
//*       A030.GEO.SUPPORT.LOADLIB                                *//
//*                                                               *//
//*****************************************************************//
//GO.STEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//*****************************************************************//
//*                                                               *//
//*  AS OF GEOSUPPPORT VERSION 10.0,                              *//
//*  DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD, ETC) *//
//*  ARE NO LONGER NEEDED AND ARE IGNORED.  GEOSUPPORT IS TAILORED *//
//*  TO USE STANDARD GEOSUPPORT DATA SET NAMES.                   *//
//*  TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.      *//
//*                                                               *//
//*****************************************************************//
//*
//GO.SYSUDUMP DD SYSOUT=A,OUTLIM=3000
//GO.SYSOUT   DD SYSOUT=A
//GO.RPTFILE  DD SYSOUT=A
//GO.INFILE  DD *
1 CHAMBERS ST                      1 HUDSON ST
1 SIXTH AVE                        1 W. 8 ST
1 DUANE ST                         1 READE ST
/*
//
```

COBOL SAMPLE PROGRAM #2 - Output Report

```
SAMPLE COBOL PROGRAM #2 EXECUTION OUTPUT

*****---------------- INPUT INTERSECTION ----------------------***** *****------------ SELECTED OUTPUT ITEMS -------------*****

B IN-STREET-NAME-1                  B IN-STREET-NAME-2                  ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES
- -------------------------------- - -------------------------------- ----- -- -------- ------- --------------------------------

1 CHAMBERS ST                       1 HUDSON ST                        10007 01 001      02      CHAMBERS STREET
                                                                                                 HUDSON STREET
                                                                                                 WEST BROADWAY

1 SIXTH AVE                         1 W. 8 ST                          10014 02 006      02      6 AVENUE
                                                                                                 GREENWICH AVENUE
                                                                                                 WEST 8 STREET

***** FUNCTION 2 GRC = 62 REASON CODE =  . READE STREET & DUANE STREET DO NOT INTERSECT
1 DUANE ST                          1 READE ST
```

ASSEMBLER SAMPLE PROGRAM #1

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - MSW

```
//ASMF1SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//**  ASSEMBLER SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1 **
//**                          MSW FORMAT                            **
//*********************************************************************
//STEP1  EXEC  ASMACLG,
//         PARM.ASM='OBJECT,NODECK',
//         PARM.LKED='XREF,LET,LIST,NCAL'
//ASM.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//           DD DSN=SYS1.MACLIB,DISP=SHR
//ASM.SYSIN  DD *
ASMF1SRC TITLE 'SAMPLE GEOSUPPORT ASSEMBLER PROGRAM 1 - MSW FORMAT'
ASMF1SRC CSECT
*****************************************************************
* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING *
* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.*
* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         *
*       ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET NAMES *
*       WOULD HAVE BEEN RETURNED BY FUNCTION 1.                 *
*****************************************************************
*
**********************************************************************
*                                                                    *
*  This program will do the following:                               *
*                                                                    *
*     * Read an instream record containing a house number,           *
*       street name, and borough code                                *
*                                                                    *
*     * Build Work Area 1 for a Function 1 call                      *
*                                                                    *
*     * Call Function 1                                              *
*                                                                    *
*     * Get ZIP code, community district, police precinct, school    *
*       district, and lists of street codes of streets intersecting  *
*       at the low and high ends of the input street address's block. *
*       These will be displayed along with the input address which   *
*       consists of borough code, house number, and street name.     *
*                                                                    *
*     * Call Function D to get the street names of the first         *
*       intersecting street on both low and high ends.               *
*                                                                    *
*     * Print the information                                        *
*                                                                    *
* NOTE that after each Geosupport call, the Return Code is checked.  *
*      If it is greater than 01, an error message is printed, and    *
*      the next input record, if any, is read.                       *
*      If it is 01, a warning message is printed, the input record is *
*      processed, and the next record is read.                       *
*      If it is zero, the input record is processed, and the next    *
*      record is read.                                               *
*                                                                    *
**********************************************************************
         SPACE
```

```
         STM   R14,R12,12(R13)   Save caller's registers
         LR    R3,R15
         LA    R12,4095(,R3)     (second base register
         LA    R12,1(,R12)        to accomodate Work Areas 1 and 2)
         USING ASMF1SRC,R3,R12
* Chain save areas
         LA    R4,MYSAVE
         ST    R13,4(,R4)    Save caller's savearea address
         ST    R4,8(,R13)    Save pgm's savearea adr in caller savearea
         LR    R13,R4        Ensure that R13 points to pgm's savearea
         SPACE 2
         XR    R15,R15                 (set OS return code to zero)
* Open input and output files
         OPEN  (INFILE,,OUTFILE,(OUTPUT))
         TM    INFILE+48,X'10'   Did input file open successfully?
         BNO   INOPNERR             (no..)
         TM    OUTFILE+48,X'10'  Did output file open successfully?
         BNO   OUTOPNER             (no..)
* Print page and report header lines
         SPACE
         PUT   OUTFILE,HDR1
         PUT   OUTFILE,HDR2
         PUT   OUTFILE,HDR3
         PUT   OUTFILE,HDR4
         B     NEXTREC
         SPACE 2
         TITLE 'READ IN-STREAM INPUT AND PREPARE FUNCTION 1 CALL'
* Read (next) input record
NEXTREC  DS    0H
         GET   INFILE,INREC
* Move input data to output record for display
         MVC   DBORO,INBORO                   borough code
         MVC   DHSE(L'W1IHSE#),INHOUSE        house number
         MVC   DSTRT,INSTREET                 street name
         SPACE
*****************************************************************
* TO MAKE A FUNCTION 1 CALL:                                    *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                         *
*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                      *
*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD      *
*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER *
*       FIELD                                                   *
*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  *
*   (6) CALL GBI WITH 2 WORKAREAS                               *
*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS               *
*****************************************************************
* Clear WA1 to blanks
         LA    R8,W1BAL    "To" address for MVCL
         LA    R9,W1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
         SPACE
* Prime Work Area 1 for Function 1 call
         MVC   W1IFUNC,=CL2'1 '              Get function code
```

```
         MVC   W1IBORO1,INBORO                    borough code
         MVC   W1IHSE#(L'W1IHSE#),INHOUSE         house number
         MVC   W1ISTRT1,INSTREET                  street name
         MVC   W1ISNL(L'W1ISNL),=C'25'    Normalized street name length
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        As of Geosupport Version 10.1,                               *
*        to receive roadbed-specific information,                     *
*        set the Roadbed Request Switch to 'R', as follows:           *
*        MVC   W1IRBRQS,C'R'                                          *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
* Call Function 1 (2-Work-Area call)
         CALL  GBI,(W1BAL,W2BAL),VL
* Check Return code
         CLC   W1ORC(2),=C'00'         Good return?
         BE    PROCESS                     Yes, process returned data
* Handle errors and warnings
ERREXIT  DS    0H
         CLC   W1ORC(2),=C'01'         Warning condition?
         BE    PUTWARN                    Yes, process warning
*                                              and then process input;
*                                         otherwise, process error
         MVC   ERINPUT,DSPLYIN    Boro code, hse no., street name
         MVC   ERFUNC,W1IFUNC              function code
         MVC   ERRET(L'W1ORC),W1ORC        return code
         MVC   ERREAS(L'W1OREASN),W1OREASN reason code
         PUT   OUTFILE,ERR1            Print error messages 1
         B     PUTMSG                                   and 2
PUTWARN  DS    0H
         MVC   WRINPUT,DSPLYIN    Boro code, hse no., street name
         MVC   WRFUNC,W1IFUNC              function code
         MVC   WRRET(L'W1ORC),W1ORC        return code
         MVC   WRREAS(L'W1OREASN),W1OREASN reason code
         PUT   OUTFILE,WRN1            Print warning messages 1
PUTMSG   DS    0H                                         and 2
         MVC   ERRWRN(L'W1OERROR),W1OERROR
         PUT   OUTFILE,ERRWRN2         Print error/warning message 2
         CLC   W1ORC(2),=C'01'         Warning condition?
         BNE   NEXTREC                  No, get next record, if any
         MVI   OINPUT,C' '              Yes,
         MVC   OINPUT+1(L'OINPUT-1),OINPUT  ensure input NOT displayd
         MVI   OUTVALID,C' '     ensure single-spacing after warning
         B     GETZIP            and continue normal processing
         SPACE
* Handle successful Geosupport calls (Return Code <= 01)
PROCESS  DS    0H
         MVC   OINPUT,DSPLYIN    Boro code, hse no., street name
         MVI   OUTVALID,C'0'     Ensure double-spacing
GETZIP   DS    0H
         MVC   OZIP,W2F1ZIP      ZIP code
         MVC   OCOMM,W2F1CDN     community district number
         MVC   OPCT,W2F1POP      police precinct
         MVC   OSCHL,W2F1SCH     school district
***** At this point, clear WA1 again, call Function D, and move ******
***** its reported Low and High Instersecting Street Names to output *
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
         TITLE 'GET LOW, HIGH INTERSECTING STREETS, USING FUNCTION D'
         SPACE 2
*************************************************************
* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND   *
* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE     *
* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS      *
* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D:  *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                     *
*   (2) SET WA1'S FUNCTION CODE FIELD TO D                  *
*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED       *
*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE      *
*              HAS SPACE FOR ONLY 25 CHARACTERS)            *
*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN       *
*       STREET NAMES FORMATTED FOR DISPLAY                  *
*   (5) MOVE WA2'S LOW PBSC FIELD TO WA1'S INPUT STREET     *
*       CODE 1 FIELD                                        *
*   (6) MOVE WA2'S HIGH PBSC FIELD TO WA1'S INPUT STREET    *
*       CODE 2 FIELD                                        *
*   (7) CALL GBI WITH 1 WORKAREA                            *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS           *
*************************************************************
         SPACE
* Clear WA1 to blanks
         LA    R8,W1BAL    "To" address for MVCL
         LA    R9,W1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
         MVC   W1ICDE1,W2F1CDEL
         MVC   W1ICDE2,W2F1CDEH
         MVC   W1IFUNC(2),=CL2'D '
         MVC   W1ISNL(L'W1ISNL),=C'25'    normalized street name length
         MVI   W1ICMPCT,C'C'              streets to be compacted
         CALL  GBI,W1BAL,VL   Call Function D
* Check Return code
         CLC   W1ORC(2),=C'00'         Good return?
         BNE   ERREXIT                     No, error or warning
*                                          Yes, complete the record
*                                               and write it out
PUTREC   DS    0H
         MVC   OLOSTRT,W1OSTRT1
         MVC   OHISTRT,W1OSTRT2
* Print an output record and get the next input record, if any
         PUT   OUTFILE,OUTVALID
         B     NEXTREC
EXIT     DS    0H
OUTOPNER DS    0H
         CLOSE (INFILE)
         TM    OUTFILE+48,X'10'   Did OUTFILE open successfully?
         BNO   INOPNERR              No, bypass closing it
         CLOSE (OUTFILE)
INOPNERR DS    0H
         L     R13,4(,R13)
         L     R14,12(,R13)
         LM    R0,R12,20(R13)
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
         BR    R14
         SPACE 2
PARAMERR DS    0H       parameter error, missing or invalid
         LA    R15,8    rc=8
         B     EXIT
         TITLE 'DATA SECTION - REGISTER ASSIGNMENTS'
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'FILE AND RECORD DEFINITIONS'
         PUSH  PRINT
         PRINT NOGEN
INFILE   DCB   DSORG=PS,MACRF=(GM),DDNAME=INFILE,                      *
               RECFM=FB,LRECL=80,BLKSIZE=400,EODAD=EXIT
         SPACE
OUTFILE  DCB   DSORG=PS,MACRF=(PM),DDNAME=SYSPRINT,                    *
               RECFM=FBA,LRECL=133,BLKSIZE=1330
*
         POP PRINT
         SPACE
INREC    DS    0CL80              Input record
INBORO   DS    CL1                    Borough code
INHOUSE  DS    CL12                   House number
INSTREET DS    CL32                   Street name
         DC    35C' '                 filler
         SPACE
* Output records: error, warning, and normal
ERR1     DS    0CL133
         DC    C'0'
ERINPUT  DS    CL48
         DC    C'*** FUNCTION '
ERFUNC   DS    CL2
         DC    C' GRC = '
ERRET    DS    CL2
         DC    C' REASON CODE = '
ERREAS   DS    CL1
         DC    CL(133-89)' '
         SPACE
WRN1     DS    0CL133
         DC    C'0'
WRINPUT  DS    CL48
         DC    C'*** FUNCTION '
WRFUNC   DS    CL2
         DC    C' WARNING, GRC = '
WRRET    DS    CL2
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
         DC    C' REASON CODE = '
WRREAS   DS    CL1
         DC    CL(133-98)' '
         SPACE
ERRWRN2  DS    0CL133
         DC    C' '
         DC    48C' '          Boro Code, House Number, Street Name
         DC    CL4'*** '
ERRWRN   DS    CL80            Error/Warning message
         SPACE
HDR1     DC    CL133'1SAMPLE ASSEMBLER #1 EXECUTION OUTPUT             *
                   '
HDR2     DC    CL133'0*****---------- INPUT ADDRESS -----------***** *C
               ****-------------------- SELECTED OUTPUT ITEMS --------C
               ---------------------*****'
HDR3     DC    CL133'0B HOUSE NUMBER IN-STREET-NAME                    *
                ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET          HIGH *
               CROSS STREET              '
HDR4     DC    CL133' - ------------ ------------------------------- -*
               ---- -- -------- ------- ------------------------ -----*
               --------------------------'
OUTVALID DS    0CL133
* Borough code, house number, and street name are from input record
         DC    C'0'
OINPUT   DS    CL48
OZIP     DS    CL5
         DC    C' '
OCOMM    DS    CL2
         DC    C' '
OPCT     DS    CL3
         DC    6C' '
OSCHL    DS    CL2
         DC    6C' '
OLOSTRT  DS    CL25  Normalized name of intersecting street at low end
         DC    C' '
OHISTRT  DS    CL25  Normalized name of intersecting street at high end
         DC    7C' '
         TITLE 'WORKING VARIABLES, VALUES, ETC.'
MYSAVE   DC    18F'0'
***********************************************************************
***** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***
***** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                      ***
***********************************************************************
         COPY  W1BAL              COPY WORK AREA 1
         EJECT
         COPY  W2BAL              COPY WORK AREA 2
         EJECT
         SPACE 2
DSPLYIN  DS    0CL48
DBORO    DS    CL1
         DC    C' '
DHSE     DS    CL12
         DC    C' '
DSTRT    DS    CL32
         DC    C' '
         SPACE 2
         TITLE 'CONSTANTS AND LITERAL POOL'
         SPACE 2
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
         LTORG
         END   ASMF1SRC
//LKED.SYSIN  DD  *
 INCLUDE INCLIB(GBI)
/*
//LKED.INCLIB   DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*                                                            *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                            *//
//*  AS OF GEOSUPPORT VERSION 10.0,                            *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP   *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:      *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                        *//
//*       A030.GEO.SUPPORT.LOADLIB                             *//
//*                                                            *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//GO.STEPLIB  DD  DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//*           DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                            *//
//*   AS OF GEOSUPPORT VERSION 10.0,                           *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,  *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT     *//
//*   IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.    *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER    *//
//*                                                            *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//SYSUDUMP DD  SYSOUT=*,OUTLIM=2000
//SYSPRINT DD  SYSOUT=*
//INFILE   DD  *
122          READE ST
1500         DUANE ST
12-4         BROADWAY
4165-100     BAISLEY BLVD
4165-1000    BAISLEY BLVD
/*
//
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - COW

```
//ASMC1SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//**  ASSEMBLER SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1 **
//**                       COW FORMAT                                **
//*********************************************************************
//STEP1  EXEC  ASMACLG,
//         PARM.ASM='OBJECT,NODECK',
//         PARM.LKED='XREF,LET,LIST,NCAL'
//ASM.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//           DD DSN=SYS1.MACLIB,DISP=SHR
//ASM.SYSIN  DD *
ASMC1SRC TITLE 'SAMPLE GEOSUPPORT ASSEMBLER PROGRAM 1 - COW FORMAT'
ASMC1SRC CSECT
*****************************************************************
* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING *
* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.*
* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         *
*       ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET NAMES *
*       WOULD HAVE BEEN RETURNED BY FUNCTION 1.                 *
*****************************************************************
*
**********************************************************************
*                                                                    *
*  This program will do the following:                               *
*                                                                    *
*     * Read an instream record containing a house number,           *
*       street name, and borough code                                *
*                                                                    *
*     * Build Work Area 1 for a Function 1 call                      *
*                                                                    *
*     * Call Function 1                                              *
*                                                                    *
*     * Get ZIP code, community district, police precinct, school    *
*       district, and lists of street codes of streets intersecting  *
*       at the low and high ends of the input street address's block. *
*       These will be displayed along with the input address which   *
*       consists of borough code, house number, and street name.     *
*                                                                    *
*     * Call Function D to get the street names of the first         *
*       intersecting street on both low and high ends.               *
*                                                                    *
*     * Print the information                                        *
*                                                                    *
* NOTE that after each Geosupport call, the Return Code is checked.  *
*      If it is greater than 01, an error message is printed, and    *
*      the next input record, if any, is read.                       *
*      If it is 01, a warning message is printed, the input record is *
*      processed, and the next record is read.                       *
*      If it is zero, the input record is processed, and the next    *
*      record is read.                                               *
*                                                                    *
**********************************************************************
         SPACE
```

```
         STM   R14,R12,12(R13)   Save caller's registers
         LR    R3,R15
         LA    R12,4095(,R3)     (second base register
         LA    R12,1(,R12)        to accomodate Work Areas 1 and 2)
         USING ASMC1SRC,R3,R12
* Chain save areas
         LA    R4,MYSAVE
         ST    R13,4(,R4)    Save caller's savearea address
         ST    R4,8(,R13)    Save pgm's savearea adr in caller savearea
         LR    R13,R4        Ensure that R13 points to pgm's savearea
         SPACE 2
         XR    R15,R15                 (set OS return code to zero)
* Open input and output files
         OPEN  (INFILE,,OUTFILE,(OUTPUT))
         TM    INFILE+48,X'10'   Did input file open successfully?
         BNO   INOPNERR             (no..)
         TM    OUTFILE+48,X'10'  Did output file open successfully?
         BNO   OUTOPNER             (no..)
* Print page and report header lines
         SPACE
         PUT   OUTFILE,HDR1
         PUT   OUTFILE,HDR2
         PUT   OUTFILE,HDR3
         PUT   OUTFILE,HDR4
         B     NEXTREC
         SPACE 2
         TITLE 'READ IN-STREAM INPUT AND PREPARE FUNCTION 1 CALL'
* Read (next) input record
NEXTREC  DS    0H
         GET   INFILE,INREC
* Move input data to output record for display
         MVC   DBORO,INBORO                     borough code
         MVC   DHSE(L'INHOUSE),INHOUSE          house number
         MVC   DSTRT,INSTREET                   street name
         SPACE
*****************************************************************
* TO MAKE A FUNCTION 1 CALL:                                    *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                         *
*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                      *
*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD      *
*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER *
*       FIELD                                                   *
*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  *
*   (6) CALL GBI WITH 2 WORKAREAS                               *
*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS               *
*****************************************************************
* Clear WA1 to blanks
         LA    R8,P1BAL    "To" address for MVCL
         LA    R9,P1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
         SPACE
* Prime Work Area 1 for Function 1 call
         MVI   P1IPLIND,C'C'                Set Work Area Format to COW
```

```
         MVC   P1IFUNC,=CL2'1 '              Get function code
         MVC   P1IBORO1,INBORO                   borough code
*
* Note COW - MSW: Display House # - P1IHSE# is a 16-byte field
*                                  W1IHSE# is a 12-byte field
*
         MVC   P1IHSE#(L'INHOUSE),INHOUSE        house number
         MVC   P1ISTRT1,INSTREET                 street name
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        As of Geosupport Version 10.1,                               *
*        to receive roadbed-specific information,                     *
*        set the Roadbed Request Switch to 'R', as follows:           *
*        MVC   P1IRBRQS,C'R'                                          *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
         MVC   P1ISNL(L'P1ISNL),=C'25'    Normalized street name length
* Call Function 1 (2-Work-Area call)
         CALL  GBI,(P1BAL,P2BAL),VL
* Check Return code
         CLC   P1ORC(2),=C'00'         Good return?
         BE    PROCESS                     Yes, process returned data
* Handle errors and warnings
ERREXIT  DS    0H
         CLC   P1ORC(2),=C'01'         Warning condition?
         BE    PUTWARN                    Yes, process warning
*                                              and then process input;
*                                         otherwise, process error
         MVC   ERINPUT,DSPLYIN    Boro code, hse no., street name
         MVC   ERFUNC,P1IFUNC              function code
         MVC   ERRET(L'P1ORC),P1ORC        return code
         MVC   ERREAS(L'P1OREASN),P1OREASN reason code
         PUT   OUTFILE,ERR1            Print error messages 1
         B     PUTMSG                                   and 2
PUTWARN  DS    0H
         MVC   WRINPUT,DSPLYIN    Boro code, hse no., street name
         MVC   WRFUNC,P1IFUNC              function code
         MVC   WRRET(L'P1ORC),P1ORC        return code
         MVC   WRREAS(L'P1OREASN),P1OREASN reason code
         PUT   OUTFILE,WRN1            Print warning messages 1
PUTMSG   DS    0H                                         and 2
         MVC   ERRWRN(L'P1OERROR),P1OERROR
         PUT   OUTFILE,ERRWRN2         Print error/warning message 2
         CLC   P1ORC(2),=C'01'         Warning condition?
         BNE   NEXTREC                  No, get next record, if any
         MVI   OINPUT,C' '              Yes,
         MVC   OINPUT+1(L'OINPUT-1),OINPUT  ensure input NOT displayd
         MVI   OUTVALID,C' '     ensure single-spacing after warning
         B     GETZIP            and continue normal processing
         SPACE
* Handle successful Geosupport calls (Return Code <= 01)
PROCESS  DS    0H
         MVC   OINPUT,DSPLYIN    Boro code, hse no., street name
         MVI   OUTVALID,C'0'     Ensure double-spacing
GETZIP   DS    0H
         MVC   OZIP,P2F1ZIP      ZIP code
```

```
         MVC   OCOMM,P2F1CDN     community district number
         MVC   OPCT,P2F1POP      police precinct
         MVC   OSCHL,P2F1SCH     school district
***** At this point, clear WA1 again, call Function D, and move ******
***** its reported Low and High Instersecting Street Names to output *
         TITLE 'GET LOW, HIGH INTERSECTING STREETS, USING FUNCTION D'
         SPACE 2
*************************************************************
* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND   *
* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE     *
* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS      *
* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D:  *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                     *
*   (2) SET WA1'S FUNCTION CODE FIELD TO D                  *
*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED       *
*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE      *
*              HAS SPACE FOR ONLY 25 CHARACTERS)            *
*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN       *
*       STREET NAMES FORMATTED FOR DISPLAY                  *
*   (5) MOVE WA2'S LOW B5SC FIELD TO WA1'S INPUT STREET     *
*       CODE 1 FIELD                                        *
*   (6) MOVE WA2'S HIGH B5SC FIELD TO WA1'S INPUT STREET    *
*       CODE 2 FIELD                                        *
*   (7) CALL GBI WITH 1 WORKAREA                            *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS           *
*************************************************************
         SPACE
* Clear WA1 to blanks
         LA    R8,P1BAL    "To" address for MVCL
         LA    R9,P1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
         MVI   P1IPLIND,C'C'                Set Work Area Format to COW
*
* Note  COW - MSW: P1ICDEx is the 10-digit street code (no boro)
*                  P1IBCDx is the Boro and 10-digit street code
*                  W1ICDEx is the packed Boro and 5-digit street code
*                  P2F1CDEx is Boro and 5-digit street code list
*                  W2F1CDEx is packed Boro and 5-digit street code list
*
         MVC   P1IBCD1(6),P2F1CDEL
         MVC   P1IBCD2(6),P2F1CDEH
         MVC   P1IFUNC(2),=CL2'D '
         MVC   P1ISNL(L'P1ISNL),=C'25'    normalized street name length
         MVI   P1ICMPCT,C'C'              streets to be compacted
         CALL  GBI,P1BAL,VL   Call Function D
* Check Return code
         CLC   P1ORC(2),=C'00'         Good return?
         BNE   ERREXIT                     No, error or warning
*                                          Yes, complete the record
*                                               and write it out
PUTREC   DS    0H
         MVC   OLOSTRT,P1OSTRT1
         MVC   OHISTRT,P1OSTRT2
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
* Print an output record and get the next input record, if any
         PUT   OUTFILE,OUTVALID
         B     NEXTREC
EXIT     DS    0H
OUTOPNER DS    0H
         CLOSE (INFILE)
         TM    OUTFILE+48,X'10'   Did OUTFILE open successfully?
         BNO   INOPNERR              No, bypass closing it
         CLOSE (OUTFILE)
INOPNERR DS    0H
         L     R13,4(,R13)
         L     R14,12(,R13)
         LM    R0,R12,20(R13)
         BR    R14
         SPACE 2
PARAMERR DS    0H       parameter error, missing or invalid
         LA    R15,8    rc=8
         B     EXIT
         TITLE 'DATA SECTION - REGISTER ASSIGNMENTS'
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'FILE AND RECORD DEFINITIONS'
         PUSH  PRINT
         PRINT NOGEN
INFILE   DCB   DSORG=PS,MACRF=(GM),DDNAME=INFILE,                      *
               RECFM=FB,LRECL=80,BLKSIZE=400,EODAD=EXIT
         SPACE
OUTFILE  DCB   DSORG=PS,MACRF=(PM),DDNAME=SYSPRINT,                    *
               RECFM=FBA,LRECL=133,BLKSIZE=1330
*
         POP PRINT
         SPACE
INREC    DS    0CL80              Input record
INBORO   DS    CL1                    Borough code
INHOUSE  DS    CL12                   House number
INSTREET DS    CL32                   Street name
         DC    35C' '                 filler
         SPACE
* Output records: error, warning, and normal
ERR1     DS    0CL133
         DC    C'0'
ERINPUT  DS    CL48
         DC    C'*** FUNCTION '
ERFUNC   DS    CL2
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
         DC    C' GRC = '
ERRET    DS    CL2
         DC    C' REASON CODE = '
ERREAS   DS    CL1
         DC    CL(133-89)' '
         SPACE
WRN1     DS    0CL133
         DC    C'0'
WRINPUT  DS    CL48
         DC    C'*** FUNCTION '
WRFUNC   DS    CL2
         DC    C' WARNING, GRC = '
WRRET    DS    CL2
         DC    C' REASON CODE = '
WRREAS   DS    CL1
         DC    CL(133-98)' '
         SPACE
ERRWRN2  DS    0CL133
         DC    C' '
         DC    48C' '           Boro Code, House Number, Street Name
         DC    CL4'*** '
ERRWRN   DS    CL80            Error/Warning message
         SPACE
HDR1     DC    CL133'1SAMPLE ASSEMBLER #1 EXECUTION OUTPUT             *
                                                                       *
                                           '
HDR2     DC    CL133'0*****---------- INPUT ADDRESS ------------***** *C
               ****-------------------- SELECTED OUTPUT ITEMS --------C
               ------------------*****'
HDR3     DC    CL133'0B HOUSE NUMBER IN-STREET-NAME                    *
                ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET          HIGH *
               CROSS STREET                '
HDR4     DC    CL133' - ------------ ------------------------------- -*
               ---- -- -------- ------- ------------------------ -----*
               -------------------------'
OUTVALID DS    0CL133
* Borough code, house number, and street name are from input record
         DC    C'0'
OINPUT   DS    CL48
OZIP     DS    CL5
         DC    C' '
OCOMM    DS    CL2
         DC    C' '
OPCT     DS    CL3
         DC    6C' '
OSCHL    DS    CL2
         DC    6C' '
OLOSTRT  DS    CL25  Normalized name of intersecting street at low end
         DC    C' '
OHISTRT  DS    CL25  Normalized name of intersecting street at high end
         DC    7C' '
         TITLE 'WORKING VARIABLES, VALUES, ETC.'
MYSAVE   DC    18F'0'
***********************************************************************
***** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***
***** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                      ***
***********************************************************************
         COPY  P1BAL              COPY WORK AREA 1
```

ASSEMBLER SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
         EJECT
         COPY  P2BAL              COPY WORK AREA 2
         EJECT
         SPACE 2
DSPLYIN  DS    0CL48
DBORO    DS    CL1
         DC    C' '
DHSE     DS    CL12
         DC    C' '
DSTRT    DS    CL32
         DC    C' '
         SPACE 2
         TITLE 'CONSTANTS AND LITERAL POOL'
         SPACE 2
         LTORG
         END   ASMC1SRC
//LKED.SYSIN  DD  *
 INCLUDE INCLIB(GBI)
/*
//LKED.INCLIB   DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*                                                            *//
//*     AS OF GEOSUPPORT VERSION 10.0,                         *//
//*     GEO.SUPPORT.PDSE.LOADLIB AND GEO.SUPPORT.LOADLIB       *//
//*     ARE REQUIRED IN THE STEPLIB (OR JOBLIB) OF THE         *//
//*     GEOSUPPORT EXECUTION STEP.                             *//
//*                                                            *//
//GO.STEPLIB  DD  DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//*           DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*                                                            *//
//*   AS OF GEOSUPPORT VERSION 10.0,                           *//
//*   DD STATEMENTS ARE NO LONGER USED TO DEFINE?              *//
//*   GEOSUPPORT DATA FILES.                                   *//
//*   DD STATEMENTS ARE NO LONGER INCLUDED FOR THE             *//
//*   GEOSUPPORT FOREGROUND FILES.                             *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER   *//
//*                                                            *//
//SYSUDUMP DD  SYSOUT=*,OUTLIM=2000
//SYSPRINT DD  SYSOUT=*
//INFILE   DD  *
122         READE ST
1500        DUANE ST
12-4        BROADWAY
4165-100    BAISLEY BLVD
4165-1000   BAISLEY BLVD
/*
//
```

ASSEMBLER SAMPLE PROGRAM #1 - Output Report

```
SAMPLE ASSEMBLER #1 EXECUTION OUTPUT

*****---------- INPUT ADDRESS ------------***** *****--------------------- SELECTED OUTPUT ITEMS ------------------------------*****

B HOUSE NUMBER IN-STREET-NAME                     ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET          HIGH CROSS STREET
- ------------ -------------------------------- ----- -- -------- ------- ------------------------- --------------------------------

1 22           READE ST                         10007 01 005      02      ELK STREET                BROADWAY

1 500          DUANE ST                         *** FUNCTION 1  GRC = 42 REASON CODE =
                                                *** ADDRESS NUMBER OUT OF RANGE

1 2-4          BROADWAY                         *** FUNCTION 1  WARNING, GRC = 01 REASON CODE = 1
                                                *** ADDR NUMBER ALTERED: RANGE ASSUMED. USING DIGITS BEFORE DASH ONLY
                                                10004 01 001      02      STONE STREET              BOWLING GREEN

4 165-100      BAISLEY BLVD                     11434 12 113      28      GUY R BREWER BOULEVARD    BEDELL STREET

4 165-1000     BAISLEY BLVD                     *** FUNCTION 1  GRC = 13 REASON CODE = 2
                                                *** ADDRESS NBR 165-1000  HAS MORE THAN 3 DIGITS AFTER THE DASH.
```

A8-52

ASSEMBLER SAMPLE PROGRAM #2

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW

```
//ASMF2SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*******************************************************************
//**  ASSEMBLER SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2 **
//**                         MSW FORMAT                             **
//*******************************************************************
//STEP1  EXEC  ASMACLG,
//         PARM.ASM='OBJECT,NODECK',
//         PARM.LKED='XREF,LET,LIST,NCAL'
//ASM.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//           DD DSN=SYS1.MACLIB,DISP=SHR
//ASM.SYSIN  DD *
ASMF2SRC TITLE 'SAMPLE GEOSUPPORT ASSEMBLER PROGRAM 2 - MSW FORMAT'
ASMF2SRC CSECT
*****************************************************************
* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING  *
* TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.   *
* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION. *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  *
* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE          *
*       ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET NAMES  *
*       WOULD HAVE BEEN RETURNED BY FUNCTION 2.                  *
*****************************************************************
*
*********************************************************************
*                                                                   *
*  This program will do the following:                              *
*     * Read an instream record containing 2 borough codes          *
*       and two street names                                        *
*                                                                   *
*     * Build Work Area 1 for a Function 2 call                     *
*                                                                   *
*     * Call Function 2                                             *
*                                                                   *
*     * Get ZIP code, community district, police precinct, school   *
*       district, and lists of intersecting street codes.           *
*       These will be displayed along with the input intersection   *
*       which consists of 2 borough codes, and 2 street names.      *
*                                                                   *
*     * Call Function D to get the street names of all intersecting *
*       streets.                                                    *
*                                                                   *
*      If it is greater than 01, an error message is printed, and   *
*      the next input record, if any, is read.                      *
*      If it is 01, a warning message is printed, the input record is *
*      processed, and the next record is read.                      *
*      If it is zero, the input record is processed, and the next   *
*      record is read.                                              *
*                                                                   *
*********************************************************************
         SPACE
         STM   R14,R12,12(R13)   Save caller's registers
         LR    R3,R15
         LA    R12,4095(,R3)     (second base register
         LA    R12,1(,R12)        to accomodate Work Areas 1 and 2)
         USING ASMF2SRC,R3,R12
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
* Chain save areas
         LA    R4,MYSAVE
         ST    R13,4(,R4)    Save caller's savearea address
         ST    R4,8(,R13)    Save pgm's savearea adr in caller savearea
         LR    R13,R4        Ensure that R13 points to pgm's savearea
         SPACE 2
         XR    R15,R15                 (set OS return code to zero)
* Open input and output files
         OPEN  (INFILE,,OUTFILE,(OUTPUT))
         TM    INFILE+48,X'10'   Did input file open successfully?
         BNO   INOPNERR             (no..)
         TM    OUTFILE+48,X'10'  Did output file open successfully?
         BNO   OUTOPNER             (no..)
* Print report header lines
         SPACE
         PUT   OUTFILE,HDR1
         PUT   OUTFILE,HDR2
         PUT   OUTFILE,HDR3
         PUT   OUTFILE,HDR4
         B     NEXTREC
         SPACE 2
         TITLE 'READ IN-STREAM INPUT AND PREPARE FUNCTION 2 CALL'
* Read (next) input record
NEXTREC  DS    0H
         GET   INFILE,INREC
* Move input data to output record for display
         MVC   DBORO1,INBORO1          First borough code
         MVC   DSTRT1,INSTRT1          First street name
         MVC   DBORO2,INBORO2          Second borough code
         MVC   DSTRT2,INSTRT2          Second street name
         SPACE
******************************************************************
* TO MAKE A FUNCTION 2 CALL:                                     *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                          *
*   (2) SET WA1'S FUNCTION CODE FIELD TO 2                       *
*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   *
*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     *
*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD *
*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   *
*   (7) CALL GBI WITH 2 WORKAREAS                                *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS                *
******************************************************************
* Clear WA1 to blanks
         LA    R8,W1BAL    "To" address for MVCL
         LA    R9,W1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
         SPACE
* Prime Work Area 1 for Function 2 call
         MVC   W1IFUNC,=CL2'2 '            Get function code
         MVC   W1IBORO1,INBORO1                borough code 1
         MVC   W1ISTRT1,INSTRT1                street name  1
         MVC   W1IBORO2,INBORO2                borough code 2
         MVC   W1ISTRT2,INSTRT2                street name  2
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
         MVC   W1ISNL(L'W1ISNL),=C'25'    Normalized street name length
* Call Function 2 (2-Work-Area call)
         CALL  GBI,(W1BAL,W2BAL),VL
* Check Return code
         CLC   W1ORC(2),=C'00'         Good return?
         BE    PROCESS                     Yes, process returned data
* Handle errors and warnings
ERREXIT  DS    0H
         CLC   W1ORC(2),=C'01'         Warning condition?
         BE    PUTWARN                    Yes, process warning
*                                              and then process input;
*                                         otherwise, process error
         MVC   ERFUNC,W1IFUNC              function code
         MVC   ERRET(L'W1ORC),W1ORC        return code
         MVC   ERREAS(L'W1OREASN),W1OREASN reason code
         MVC   ERRMSG(L'W1OERROR),W1OERROR Geosupport error message
         PUT   OUTFILE,ERR1            Print error message 1
         MVC   ERINPUT,DSPLYIN    2 boro codes and 2 street names
         PUT   OUTFILE,ERR2            Print error message 2
         B     NEXTREC
PUTWARN  DS    0H
         MVC   WRFUNC,W1IFUNC              function code
         MVC   WRRET(L'W1ORC),W1ORC        return code
         MVC   WRREAS(L'W1OREASN),W1OREASN reason code
         MVC   WRNMSG(L'W1OERROR),W1OERROR Geosupport warning message
         PUT   OUTFILE,WARN            Print warning message
         SPACE
* Handle successful Geosupport calls (Return Code <= 01)
PROCESS  DS    0H
         MVI   OUTFIXED,C'0'      Init. carriage control to dbl-space
         CLC   W1ORC(2),=C'01'    Was a warning issued?
         BNE   MOVEOUT               No..
         MVI   OUTFIXED,C' '         Yes, single-space output instead
MOVEOUT  DS    0H
         MVC   OINPUT,DSPLYIN    Pair of boro codes and street names
         MVC   OZIP,W2F2ZIP      ZIP code
         MVC   OCOMM,W2F2CDN     community district number
         MVC   OPCT,W2F2POP      police precinct
         MVC   OSCHL,W2F2SCH     school district
***** At this point, clear WA1 again, call Function D, and move ******
***** all reported Intersecting Street Names to output          ******
         TITLE 'GET INTERSECTING STREET NAMES, USING FUNCTION D'
         SPACE 2
*************************************************************
* TO GET THE STREET NAMES FOR INTERSECTING STREET CODES     *
* MAKE A FUNCTION D CALL:                                   *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                     *
*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D              *
*   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN       *
*       STREET NAMES FORMATTED FOR DISPLAY                  *
*   (4) MOVE THE PACKED BORO AND STREET CODE TO             *
*       WA1'S INPUT STREET CODE 1 FIELD                     *
*   (5) CALL GBI WITH 1 WORKAREA                            *
*   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS           *
*************************************************************
         SPACE
* For each street code of intersecting streets, including those input,
* call Function D to get the corresponding street name
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
         XR    R4,R4
         MVC   INTWK,W2F2#INT   get count of intersecting streets.
         NI    INTWK,X'0F'      remove zone, leaving numeric
         IC    R4,INTWK         count of intersecting streets.
         LA    R5,W2F2CODE      point to street code(s).
         SPACE
INTRLOOP DS    0H
* Clear WA1 to blanks
         LA    R8,W1BAL    "To" address for MVCL
         LA    R9,W1LENGTH     "To" length
         XR    R11,R11        for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
         SPACE
         MVC   W1IFUNC(2),=CL2'D '
         MVC   W1ISNL(L'W1ISNL),=C'25'    normalized street name length
         MVI   W1ICMPCT,C'C'              streets to be compacted
         MVC   W1ICDE1(L'W1ICDE1),0(R5)   Intersecting street code
         CALL  GBI,W1BAL,VL   Call Function D
* Check Return code
         CLC   W1ORC(2),=C'00'         Good return?
         BNE   ERREXIT                     No, error or warning
*                                          Yes, complete the record
*                                               and write it out
PFIX     NOP   PVAR
         OI    PFIX+1,X'F0'
         MVC   OINTRSC1,W1OSTRT1
*  put out the initial output including the first intersecting street
         PUT   OUTFILE,OUTFIXED
         B     NEXTSC     Now get the rest of the street codes, if any
PVAR     DS    0H
         MVC   OINTRSCN,W1OSTRT1
* Print an output record and get the next intersecting street, if any
         PUT   OUTFILE,OUTVAR
NEXTSC   DS    0H
         LA    R5,4(,R5)    point to next intersecting street code
         BCT   R4,INTRLOOP  if any, and process it;
         NI    PFIX+1,X'0F'    reset 1st-time (fixed/variable) switch
         B     NEXTREC      then, process next input record, if any
         SPACE
EXIT     DS    0H
OUTOPNER DS    0H
         CLOSE (INFILE)
         TM    OUTFILE+48,X'10'   Did OUTFILE open successfully?
         BNO   INOPNERR              No, bypass closing it
         CLOSE (OUTFILE)
INOPNERR DS    0H
         L     R13,4(,R13)
         L     R14,12(,R13)
         LM    R0,R12,20(R13)
         BR    R14
         SPACE 2
PARAMERR DS    0H       parameter error, missing or invalid
         LA    R15,8    rc=8
         B     EXIT
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
         TITLE 'DATA SECTION - REGISTER ASSIGNMENTS'
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'FILE AND RECORD DEFINITIONS'
         PUSH  PRINT
         PRINT NOGEN
INFILE   DCB   DSORG=PS,MACRF=(GM),DDNAME=INFILE,                      *
               RECFM=FB,LRECL=80,BLKSIZE=400,EODAD=EXIT
         SPACE
OUTFILE  DCB   DSORG=PS,MACRF=(PM),DDNAME=SYSPRINT,                    *
               RECFM=FBA,LRECL=133,BLKSIZE=1330
*
         POP PRINT
         SPACE
INREC    DS    0CL80              Input record
INBORO1  DS    CL1                    First borough code
INSTRT1  DS    CL32                   First street name
INBORO2  DS    CL1                    Second borough code
INSTRT2  DS    CL32                   Second street name
         DC    14C' '                 filler
         SPACE
* Output records: header, normal, warning, and error
         SPACE
* header records
HDR1     DC    CL133'1SAMPLE ASSEMBLER #2 EXECUTION OUTPUT             *
                                                                       *
                           '
HDR2     DC    CL133'0*****----------------- INPUT INTERSECTION -------C
               --------------***** *****--------------- SELECTED OUTPUT*
                ITEMS ---------------*****'
HDR3     DC    CL133'0B IN-STREET-NAME-1                 B IN-STREET-NA*
               ME-2                  ZIP CD NYPD-PCT SCHLDST INTERSECTI*
               NG STREET NAMES            '
HDR4     DC    CL133' - -------------------------------- - ------------*
               ------------------ ----- -- -------- ------- -----------*
               ---------------------------'
         SPACE
* normal records, i.e., output for valid data
OUTFIXED DS    0CL133                 Fixed output
* Borough codes and street names for each of 2 streets are from input
         DC    C'0'
OINPUT   DS    CL69
OZIP     DS    CL5
         DC    C' '
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
OCOMM    DS    CL2
         DC    C' '
OPCT     DS    CL3
         DC    6C' '
OSCHL    DS    CL2
         DC    6C' '
OINTRSC1 DS    CL25  Normalized name of first intersecting street
         SPACE
OUTVAR   DS    0CL133 Output line repeated per No. of Intersecting Sts.
         DC    C' '
         DC    95C' '
OINTRSCN DS    CL25  Normalized name of additional intersecting street
         DC    (133-121)C' '
         SPACE
* warning record
WARN     DS    0CL133
         DC    C'0'
         DC    C'**** FUNCTION '
WRFUNC   DS    CL2
         DC    C' WARNING, GRC = '
WRRET    DS    CL2
         DC    C' REASON CODE = '
WRREAS   DS    CL1
         DC    C'. '
WRNMSG   DS    CL80            Warning message
         SPACE
* error records
ERR1     DS    0CL133
         DC    C'0'
         DC    C'***** FUNCTION '
ERFUNC   DS    CL2
         DC    C' GRC = '
ERRET    DS    CL2
         DC    C' REASON = '
ERREAS   DS    CL1
         DC    C'. '
ERRMSG   DS    CL80            Error message
         DC    CL(133-120)' '
         SPACE
ERR2     DS    0CL133
         DC    C' '
ERINPUT  DS    CL69
         DC    CL(133-70)' '
         TITLE 'WORKING VARIABLES, VALUES, ETC.'
#INTER   DS    D         Working field for no. of intersecting streets
MYSAVE   DC    18F'0'
***********************************************************************
***** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***
***** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                      ***
***********************************************************************
         COPY  W1BAL              COPY WORK AREA 1
         EJECT
         COPY  W2BAL              COPY WORK AREA 2
         EJECT
         SPACE 2
DSPLYIN  DS    0CL69
DBORO1   DS    CL1
         DC    C' '
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
DSTRT1   DS    CL32
         DC    C' '
DBORO2   DS    CL1
         DC    C' '
DSTRT2   DS    CL32
         SPACE 2
INTWK    DS    XL1      work field for number of intersecting streets
         TITLE 'CONSTANTS AND LITERAL POOL'
         SPACE 2
         LTORG
         END   ASMF2SRC
//LKED.SYSIN  DD  *
 INCLUDE INCLIB(GBI)
/*
//LKED.INCLIB   DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*                                                              *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                              *//
//*  AS OF GEOSUPPORT VERSION 10.0,                              *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP    *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:       *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                          *//
//*       A030.GEO.SUPPORT.LOADLIB                               *//
//*                                                              *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//GO.STEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//           DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                              *//
//*   AS OF GEOSUPPORT VERSION 10.0,                             *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,   *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT      *//
//*   IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.     *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.    *//
//*                                                              *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//SYSUDUMP DD  SYSOUT=*,OUTLIM=2000
//SYSPRINT DD  SYSOUT=*
//INFILE   DD  *
1CHAMBERS ST                    1HUDSON ST
1SIXTH AVE                      1W. 8 ST
1DUANE ST                       1READE ST
/*
//
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - COW

```
//ASMC2SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*******************************************************************
//**  ASSEMBLER SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2 **
//**                         COW FORMAT                             **
//*******************************************************************
//STEP1  EXEC  ASMACLG,
//         PARM.ASM='OBJECT,NODECK',
//         PARM.LKED='XREF,LET,LIST,NCAL'
//ASM.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//           DD DSN=SYS1.MACLIB,DISP=SHR
//ASM.SYSIN  DD *
ASMC2SRC TITLE 'SAMPLE GEOSUPPORT ASSEMBLER PROGRAM 2 - COW FORMAT'
ASMC2SRC CSECT
*****************************************************************
* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING  *
* TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.   *
* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION. *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  *
* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE          *
*       ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET NAMES  *
*       WOULD HAVE BEEN RETURNED BY FUNCTION 2.                  *
*****************************************************************
*
**********************************************************************
*                                                                    *
*  This program will do the following:                               *
*     * Read an instream record containing 2 borough codes           *
*       and two street names                                         *
*                                                                    *
*     * Build Work Area 1 for a Function 2 call                      *
*                                                                    *
*     * Call Function 2                                              *
*                                                                    *
*     * Get ZIP code, community district, police precinct, school    *
*       district, and lists of intersecting street codes.            *
*       These will be displayed along with the input intersection    *
*       which consists of 2 borough codes, and 2 street names.       *
*                                                                    *
*     * Call Function D to get the street names of all intersecting  *
*       streets.                                                     *
*                                                                    *
*      If it is greater than 01, an error message is printed, and    *
*      the next input record, if any, is read.                       *
*      If it is 01, a warning message is printed, the input record is *
*      processed, and the next record is read.                       *
*      If it is zero, the input record is processed, and the next    *
*      record is read.                                               *
*                                                                    *
**********************************************************************
         SPACE
         STM   R14,R12,12(R13)   Save caller's registers
         LR    R3,R15
         LA    R12,4095(,R3)     (second base register
         LA    R12,1(,R12)        to accomodate Work Areas 1 and 2)
         USING ASMC2SRC,R3,R12
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
* Chain save areas
         LA    R4,MYSAVE
         ST    R13,4(,R4)    Save caller's savearea address
         ST    R4,8(,R13)    Save pgm's savearea adr in caller savearea
         LR    R13,R4        Ensure that R13 points to pgm's savearea
         SPACE 2
         XR    R15,R15                 (set OS return code to zero)
* Open input and output files
         OPEN  (INFILE,,OUTFILE,(OUTPUT))
         TM    INFILE+48,X'10'   Did input file open successfully?
         BNO   INOPNERR             (no..)
         TM    OUTFILE+48,X'10'  Did output file open successfully?
         BNO   OUTOPNER             (no..)
* Print report header lines
         SPACE
         PUT   OUTFILE,HDR1
         PUT   OUTFILE,HDR2
         PUT   OUTFILE,HDR3
         PUT   OUTFILE,HDR4
         B     NEXTREC
         SPACE 2
         TITLE 'READ IN-STREAM INPUT AND PREPARE FUNCTION 2 CALL'
* Read (next) input record
NEXTREC  DS    0H
         GET   INFILE,INREC
* Move input data to output record for display
         MVC   DBORO1,INBORO1          First borough code
         MVC   DSTRT1,INSTRT1          First street name
         MVC   DBORO2,INBORO2          Second borough code
         MVC   DSTRT2,INSTRT2          Second street name
         SPACE
******************************************************************
* TO MAKE A COW FORMAT FUNCTION 2 CALL:                          *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                          *
*       AND SET WORK AREA FORMAT FLAG TO 'C'                     *
*   (2) SET WA1'S FUNCTION CODE FIELD TO 2                       *
*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   *
*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     *
*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD *
*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   *
*   (7) CALL GBI WITH 2 WORKAREAS                                *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS                *
******************************************************************
* Clear WA1 to blanks
         LA    R8,P1BAL    "To" address for MVCL
         LA    R9,P1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
*
         MVI   P1IPLIND,C'C'  set work area format indicator to COW
*
         SPACE
* Prime Work Area 1 for Function 2 call
         MVC   P1IFUNC,=CL2'2 '             Get function code
```

```
         MVC   P1IBORO1,INBORO1                  borough code 1
         MVC   P1ISTRT1,INSTRT1                  street name  1
         MVC   P1IBORO2,INBORO2                  borough code 2
         MVC   P1ISTRT2,INSTRT2                  street name  2
         MVC   P1ISNL(L'P1ISNL),=C'25'    Normalized street name length
* Call Function 2 (2-Work-Area call)
         CALL  GBI,(P1BAL,P2BAL),VL
* Check Return code
         CLC   P1ORC(2),=C'00'         Good return?
         BE    PROCESS                     Yes, process returned data
* Handle errors and warnings
ERREXIT  DS    0H
         CLC   P1ORC(2),=C'01'         Warning condition?
         BE    PUTWARN                    Yes, process warning
*                                              and then process input;
*                                         otherwise, process error
         MVC   ERFUNC,P1IFUNC              function code
         MVC   ERRET(L'P1ORC),P1ORC        return code
         MVC   ERREAS(L'P1OREASN),P1OREASN reason code
         MVC   ERRMSG(L'P1OERROR),P1OERROR Geosupport error message
         PUT   OUTFILE,ERR1            Print error message 1
         MVC   ERINPUT,DSPLYIN    2 boro codes and 2 street names
         PUT   OUTFILE,ERR2            Print error message 2
         B     NEXTREC
PUTWARN  DS    0H
         MVC   WRFUNC,P1IFUNC              function code
         MVC   WRRET(L'P1ORC),P1ORC        return code
         MVC   WRREAS(L'P1OREASN),P1OREASN reason code
         MVC   WRNMSG(L'P1OERROR),P1OERROR Geosupport warning message
         PUT   OUTFILE,WARN            Print warning message
         SPACE
* Handle successful Geosupport calls (Return Code <= 01)
PROCESS  DS    0H
         MVI   OUTFIXED,C'0'      Init. carriage control to dbl-space
         CLC   P1ORC(2),=C'01'    Was a warning issued?
         BNE   MOVEOUT               No..
         MVI   OUTFIXED,C' '         Yes, single-space output instead
MOVEOUT  DS    0H
         MVC   OINPUT,DSPLYIN    Pair of boro codes and street names
         MVC   OZIP,P2F2ZIP      ZIP code
         MVC   OCOMM,P2F2CDN     community district number
         MVC   OPCT,P2F2POP      police precinct
         MVC   OSCHL,P2F2SCH     school district
***** At this point, clear WA1 again, call Function D, and move ******
***** all reported Intersecting Street Names to output          ******
         TITLE 'GET INTERSECTING STREET NAMES, USING FUNCTION D'
         SPACE 2
*************************************************************
* TO GET THE STREET NAMES FOR INTERSECTING STREET CODES     *
* MAKE A FUNCTION D CALL:                                   *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                     *
*       AND SET WORK AREA FORMAT FLAG TO 'C'                *
*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D              *
*   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN       *
*       STREET NAMES FORMATTED FOR DISPLAY                  *
*   (4) MOVE THE PACKED BORO AND STREET CODE TO             *
*       WA1'S INPUT STREET CODE 1 FIELD                     *
*   (5) CALL GBI WITH 1 WORKAREA                            *
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
*   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS          *
************************************************************
         SPACE
* For each street code of intersecting streets, including those input,
* call Function D to get the corresponding street name
         XR    R4,R4
         MVC   INTWK,P2F2#INT   get count of intersecting streets.
         NI    INTWK,X'0F'      remove zone, leaving numeric
         IC    R4,INTWK         count of intersecting streets.
         LA    R5,P2F2CODE      point to street code(s).
         SPACE
INTRLOOP DS    0H
* Clear WA1 to blanks
         LA    R8,P1BAL    "To" address for MVCL
         LA    R9,P1LENGTH     "To" length
         XR    R11,R11       for blanking out std WA1,
         ICM   R11,B'1000',=C' '  rather than moving data
* ...since if the "from" length reg. has lo-order zeroes, MVCL will
* pad the target area with the pad character of the "from" register
* and do nothing else (the "from" address register is not used).
         MVCL  R8,R10
*
         MVI   P1IPLIND,C'C'  set work area format indicator to COW
*
         SPACE
         MVC   P1IFUNC(2),=CL2'D '
         MVC   P1ISNL(L'P1ISNL),=C'25'    normalized street name length
         MVI   P1ICMPCT,C'C'              streets to be compacted
*
* Note  COW - MSW: P1ICDEx is the 10-digit street code (no boro)
*                  P1IBCDx is the Boro and 10-digit street code
*                  W1ICDEx is the packed Boro and 5-digit street code
*                  P2F2CODE is Boro and 5-digit street code list
*                  W2F2CODE is packed Boro and 5-digit street code list
*
         MVC   P1IBCD1(LB5SC),0(R5)   Intersecting boro and street code
         CALL  GBI,P1BAL,VL   Call Function D
* Check Return code
         CLC   P1ORC(2),=C'00'         Good return?
         BNE   ERREXIT                     No, error or warning
*                                          Yes, complete the record
*                                               and write it out
PFIX     NOP   PVAR
         OI    PFIX+1,X'F0'
         MVC   OINTRSC1,P1OSTRT1
*  put out the initial output including the first intersecting street
         PUT   OUTFILE,OUTFIXED
         B     NEXTSC     Now get the rest of the street codes, if any
PVAR     DS    0H
         MVC   OINTRSCN,P1OSTRT1
* Print an output record and get the next intersecting street, if any
         PUT   OUTFILE,OUTVAR
NEXTSC   DS    0H
         LA    R5,LB5SC(,R5) point to next intersecting street code
         BCT   R4,INTRLOOP  if any, and process it;
         NI    PFIX+1,X'0F'    reset 1st-time (fixed/variable) switch
         B     NEXTREC      then, process next input record, if any
         SPACE
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
EXIT     DS    0H
OUTOPNER DS    0H
         CLOSE (INFILE)
         TM    OUTFILE+48,X'10'   Did OUTFILE open successfully?
         BNO   INOPNERR              No, bypass closing it
         CLOSE (OUTFILE)
INOPNERR DS    0H
         L     R13,4(,R13)
         L     R14,12(,R13)
         LM    R0,R12,20(R13)
         BR    R14
         SPACE 2
PARAMERR DS    0H       parameter error, missing or invalid
         LA    R15,8    rc=8
         B     EXIT
         TITLE 'DATA SECTION - REGISTER ASSIGNMENTS'
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'FILE AND RECORD DEFINITIONS'
         PUSH  PRINT
         PRINT NOGEN
INFILE   DCB   DSORG=PS,MACRF=(GM),DDNAME=INFILE,                      *
               RECFM=FB,LRECL=80,BLKSIZE=400,EODAD=EXIT
         SPACE
OUTFILE  DCB   DSORG=PS,MACRF=(PM),DDNAME=SYSPRINT,                    *
               RECFM=FBA,LRECL=133,BLKSIZE=1330
*
         POP PRINT
         SPACE
INREC    DS    0CL80              Input record
INBORO1  DS    CL1                    First borough code
INSTRT1  DS    CL32                   First street name
INBORO2  DS    CL1                    Second borough code
INSTRT2  DS    CL32                   Second street name
         DC    14C' '                 filler
         SPACE
* Output records: header, normal, warning, and error
         SPACE
* header records
HDR1     DC    CL133'1SAMPLE ASSEMBLER #2 EXECUTION OUTPUT             *
                                                                       *
                                          '
HDR2     DC    CL133'0*****----------------- INPUT INTERSECTION -------C
               --------------***** *****--------------- SELECTED OUTPUT*
```

```
                 ITEMS ---------------*****'
HDR3     DC    CL133'0B IN-STREET-NAME-1                 B IN-STREET-NA*
               ME-2                   ZIP CD NYPD-PCT SCHLDST INTERSECTI*
               NG STREET NAMES            '
HDR4     DC    CL133' - ------------------------------- - -----------*
               ----------------- ----- -- -------- ------- ----------*
               --------------------------'
         SPACE
* normal records, i.e., output for valid data
OUTFIXED DS    0CL133                 Fixed output
* Borough codes and street names for each of 2 streets are from input
         DC    C'0'
OINPUT   DS    CL69
OZIP     DS    CL5
         DC    C' '
OCOMM    DS    CL2
         DC    C' '
OPCT     DS    CL3
         DC    6C' '
OSCHL    DS    CL2
         DC    6C' '
OINTRSC1 DS    CL25  Normalized name of first intersecting street
         SPACE
OUTVAR   DS    0CL133 Output line repeated per No. of Intersecting Sts.
         DC    C' '
         DC    95C' '
OINTRSCN DS    CL25  Normalized name of additional intersecting street
         DC    (133-121)C' '
         SPACE
* warning record
WARN     DS    0CL133
         DC    C'0'
         DC    C'**** FUNCTION '
WRFUNC   DS    CL2
         DC    C' WARNING, GRC = '
WRRET    DS    CL2
         DC    C' REASON CODE = '
WRREAS   DS    CL1
         DC    C'. '
WRNMSG   DS    CL80            Warning message
         SPACE
* error records
ERR1     DS    0CL133
         DC    C'0'
         DC    C'***** FUNCTION '
ERFUNC   DS    CL2
         DC    C' GRC = '
ERRET    DS    CL2
         DC    C' REASON = '
ERREAS   DS    CL1
         DC    C'. '
ERRMSG   DS    CL80            Error message
         DC    CL(133-120)' '
         SPACE
ERR2     DS    0CL133
         DC    C' '
ERINPUT  DS    CL69
         DC    CL(133-70)' '
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
         TITLE 'WORKING VARIABLES, VALUES, ETC.'
#INTER   DS    D        Working field for no. of intersecting streets
MYSAVE   DC    18F'0'
***********************************************************************
***** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***
***** COPY STATEMENTS) IS STRONGLY ENCOURAGED.                      ***
***********************************************************************
         COPY  P1BAL              COPY WORK AREA 1
         EJECT
         COPY  P2BAL              COPY WORK AREA 2
         EJECT
         SPACE 2
DSPLYIN  DS    0CL69
DBORO1   DS    CL1
         DC    C' '
DSTRT1   DS    CL32
         DC    C' '
DBORO2   DS    CL1
         DC    C' '
DSTRT2   DS    CL32
         SPACE 2
INTWK    DS    XL1      work field for number of intersecting streets
LB5SC    EQU   6        length of Boro and 5-digit street code
         TITLE 'CONSTANTS AND LITERAL POOL'
         SPACE 2
         LTORG
         END   ASMC2SRC
//LKED.SYSIN  DD  *
 INCLUDE INCLIB(GBI)
/*
//LKED.INCLIB   DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*                                                              *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                              *//
//*  AS OF GEOSUPPORT VERSION 10.0,                              *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP    *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:       *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                          *//
//*       A030.GEO.SUPPORT.LOADLIB                               *//
//*                                                              *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//GO.STEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                              *//
//*   AS OF GEOSUPPORT VERSION 10.0,                             *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,   *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT      *//
//*   IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.     *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.    *//
//*                                                              *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//SYSUDUMP DD  SYSOUT=*,OUTLIM=2000
//SYSPRINT DD  SYSOUT=*
//INFILE   DD  *
1CHAMBERS ST                     1HUDSON ST
```

ASSEMBLER SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
1SIXTH AVE                         1W. 8 ST
1DUANE ST                          1READE ST
/*
//
```

ASSEMBLER SAMPLE PROGRAM #2 - Output Report

```
SAMPLE ASSEMBLER #2 EXECUTION OUTPUT

*****---------------- INPUT INTERSECTION ---------------------***** *****--------------- SELECTED OUTPUT ITEMS ---------------*****

B IN-STREET-NAME-1                  B IN-STREET-NAME-2                  ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES
- --------------------------------- - ------------------------------ ----- -- -------- ------- -------------------------------------

1 CHAMBERS ST                       1 HUDSON ST                       10007 01 001      02      CHAMBERS STREET
                                                                                                HUDSON STREET
                                                                                                WEST BROADWAY

1 SIXTH AVE                         1 W. 8 ST                         10014 02 006      02      6 AVENUE
                                                                                                GREENWICH AVENUE
                                                                                                WEST 8 STREET

***** FUNCTION 2  GRC = 62 REASON =  . READE STREET & DUANE STREET DO NOT INTERSECT
1 DUANE ST                          1 READE ST
```

PL/1 SAMPLE PROGRAM #1

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

PL/1 SAMPLE PROGRAM #1 - Job Stream - MSW

```
//PL1F1BAT  JOB YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  PL1 SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1. ******
//***                  (MSW FORMAT)                              ******
//*********************************************************************
//STEP1      EXEC IBMZCPLG,REGION=0M,GOPGM='PL1F1SC',
//           PARM.PLI='S,GS,INCLUDE',
//           PARM.LKED='AMODE(31)'
//PLI.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//SYSIN      DD *
 PL1F1SC:    PROC OPTIONS(MAIN);
   /*****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING */
   /* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.*/
   /* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     */
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         */
   /*       ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET       */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.           */
   /*****************************************************************/

 DCL EOF                     BIT(1)  INIT('0'B),
     YES                     BIT(1)  INIT('1'B),
     NO                      BIT(1)  INIT('0'B),
     ADDR                    BUILTIN,
     (I,J)                   FIXED BIN(15) INIT(0);

 /*******************************************************************/
 /**************   GBI DECLARATION BELOW IS REQUIRED   **************/
 /*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE *****/
 /*** %INCLUDE STATEMENTS) IS STRONGLY ENCOURAGED.           *******/
 /*******************************************************************/
 DCL GBI                     ENTRY OPTIONS(ASM,INTER);
 %INCLUDE W1PL1;
 %INCLUDE W2PL1;

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
 DCL INFILE FILE STREAM INPUT;
 DCL IN_BORO                 CHAR(01),
     IN_HOUSENUM             CHAR(12),
     IN_STREET_NAME          CHAR(32);

 /******** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT *********/
 DCL SYSPRINT FILE STREAM OUTPUT PRINT;
 ON ENDPAGE(SYSPRINT)
   PUT EDIT('SAMPLE PL1 PROGRAM #1 EXECUTION OUTPUT',
      '*****---------- INPUT ADDRESS ------------***** '||
      '*****---------------------'||
      ' SELECTED OUTPUT ITEMS -----------------------*****',
      'B HOUSE NUMBER IN-STREET-NAME                   '||
      '  ZIP CD NYPD-PCT SCHLDST '||
      'LOW CROSS STREET          HIGH CROSS STREET        ',
      '_ ____________ '||(32)'_'||' '||
      '_____ __ ________ _______ '||
      (25)'_'||' '||(25)'_')
```

PL/1 SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
    (PAGE,COL(1),A,SKIP(2),COL(1),A,COL(1),A,SKIP(0),COL(1),A);
OPEN FILE(SYSPRINT) LINESIZE(133);
SIGNAL ENDPAGE(SYSPRINT);
/*****************************************************************/

ON ENDFILE(INFILE)  BEGIN; EOF=YES; GOTO ENDLOOP;  END;
OPEN FILE(INFILE);

DO WHILE (EOF = NO);

  /********** REPLACE CODE BELOW WITH YOUR OWN INPUT *************/
  GET FILE(INFILE) EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME)
                     (COL(1),A(1),X(1),A(12),X(1),A(32));

  /*****************************************************************/
  /* TO MAKE A FUNCTION 1 CALL:                                    */
  /*   (1) INITIALIZE WORKAREA 1 TO SPACES                         */
  /*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                      */
  /*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD      */
  /*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER */
  /*       FIELD                                                   */
  /*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  */
  /*   (6) CALL GBI WITH 2 WORKAREAS                               */
  /*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS               */
  /*****************************************************************/
  WORK1PL1 = ' ';
  GEO_WA1_IN_FUNCTION_1 = '1';
  GEO_WA1_IN_BORO       = IN_BORO;
  GEO_WA1_IN_HOUSENUM   = IN_HOUSENUM;
  GEO_WA1_IN_STREET_1   = IN_STREET_NAME;

  CALL GBI(W1PL1,W2PL1);

  IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 ¼= '00' &
     GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 ¼= '01'
  THEN DO;
         /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
         PUT EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME,
                  '*** FUNCTION 1 GRC =',
                  GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2,
                  'REASON CODE =',GEO_WA1_OUT_REASON_CODE,
                  '*** ',GEO_WA1_OUT_ERROR_MESSAGE)
                 (SKIP(2),COL(1),(7)(A,X(1)),SKIP(1),COL(49),A,A);
       END;
  ELSE DO;
         PUT EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME)
                 (SKIP(2),COL(1),(3)(A,X(1)));
         IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 = '01'
         THEN DO;
                /** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE **/
                PUT EDIT('*** FUNCTION 1 WARNING, GRC =',
                         GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2,
                         'REASON CODE =',GEO_WA1_OUT_REASON_CODE,
                         '*** ',GEO_WA1_OUT_ERROR_MESSAGE)
                        (COL(49),(4)(A,X(1)),SKIP(1),COL(49),A,A);
              END;
         /**************************************************/
         /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ****/
```

PL/1 SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
          /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS    ****/
          /**************************************************/
          PUT EDIT(GEO_WA2_FN1_ZIP,GEO_WA2_FN1_COMDIST_NUMBER,
                   GEO_WA2_FN1_POL_PRECINCT,GEO_WA2_FN1_SCHOOLDIST)
                  (COL(49),(3)(A,X(1)),X(5),A);
          /*********************************************************/
          /* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND */
          /* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE   */
          /* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS    */
          /* FROM THE HIGH AND LOW STREET CODE LISTS CALL            */
          /* FUNCTION D:                                             */
          /*    (1) INITIALIZE WORKAREA 1 TO SPACES                  */
          /*    (2) SET WA1'S FUNCTION CODE FIELD TO D               */
          /*    (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED    */
          /*        VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE   */
          /*               HAS SPACE FOR ONLY 25 CHARACTERS)         */
          /*    (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN    */
          /*        STREET NAMES FORMATTED FOR DISPLAY               */
          /*    (5) MOVE WA2'S LOW PBSC FIELD TO WA1'S INPUT STREET  */
          /*        CODE 1 FIELD                                     */
          /*    (6) MOVE WA2'S HIGH PBSC FIELD TO WA1'S INPUT STREET */
          /*        CODE 2 FIELD                                     */
          /*    (7) CALL GBI WITH 1 WORKAREA                         */
          /*    (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS        */
          /*********************************************************/
          WORK1PL1 = ' ';
          GEO_WA1_IN_FUNCTION_1        = 'D';
          GEO_WA1_IN_SNL               = '25';
          GEO_WA1_IN_COMPACT_NAME_FLAG = 'C';
          GEO_WA1_IN_STREETCODE_1      = GEO_WA2_FN1_LOW_PBSC(1);
          GEO_WA1_IN_STREETCODE_2      = GEO_WA2_FN1_HI_PBSC(1);
          CALL GBI(W1PL1);
          IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 = '00'
          THEN DO;
                 /******** INSERT YOUR OWN CODE HERE *******/
                 PUT EDIT(GEO_WA1_OUT_STREET_1,GEO_WA1_OUT_STREET_2)
                         (COL(75),A(25),X(1),A(25));
               END;
          ELSE DO;
                 /*** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE ***/
                 PUT EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME,
                          '*** FUNCTION D GRC =',
                          GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2,
                          'REASON CODE =',GEO_WA1_OUT_REASON_CODE,',',
                          '*** ',GEO_WA1_OUT_ERROR_MESSAGE)
                         (SKIP(2),COL(1),(8)(A,X(1)),
                          SKIP(1),COL(49),A,A);
               END;
        END;
 ENDLOOP:  END;
 CLOSE FILE(INFILE);
 END PL1F1SC;
/*
//LKED.SYSIN  DD *
  INCLUDE INCLIB(GBI)
/*
//INCLIB      DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
```

PL/1 SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                                */
//*   THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP  *//
//*   MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:     *//
//*        A030.GEO.SUPPORT.PDSE.LOADLIB                        *//
//*        A030.GEO.SUPPORT.LOADLIB                             *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//GO.STEPLIB  DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//GO.SYSLMOD  DD DSN=&&GOSET,DISP=(OLD,DELETE)
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                            *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,  *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT     *//
//*   IS TAILORD TO USE STANDARD GEOSUPPORT DATA SET NAMES.     *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.   *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//GO.SYSPRINT DD SYSOUT=A
//GO.INFILE   DD *
1 22           READE ST
1 500          DUANE ST
1 2-4          BROADWAY
4 165-100      BAISLEY BLVD
4 165-1000     BAISLEY BLVD
/*
//
```

PL/1 SAMPLE PROGRAM #1 - Job Stream - COW

```
//PL1C1SRC  JOB YOUR-JOB-CARD-INFORMATION
//*
//********************************************************************
//***  PL1 SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1. ******
//***                 (COW FORMAT)                            ******
//********************************************************************
//STEP1      EXEC IBMZCPLG,REGION=0M,GOPGM='PL1C1SC',
//           PARM.PLI='S,GS,INCLUDE',
//           PARM.LKED='AMODE(31),LIST'
//PLI.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//SYSIN      DD *
 PL1C1SC:   PROC OPTIONS(MAIN);
   /*****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING */
   /* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.*/
   /* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     */
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         */
   /*       ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET       */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.           */
   /*****************************************************************/

 DCL EOF                        BIT(1)  INIT('0'B),
     YES                        BIT(1)  INIT('1'B),
     NO                         BIT(1)  INIT('0'B),
     ADDR                       BUILTIN,
     (I,J)                      FIXED BIN(15) INIT(0);

 /*******************************************************************/
 /**************   GBI DECLARATION BELOW IS REQUIRED   **************/
 /*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE *****/
 /*** %INCLUDE STATEMENTS) IS STRONGLY ENCOURAGED.           *******/
 /*******************************************************************/
 DCL GBI                        ENTRY OPTIONS(ASM,INTER);
 %INCLUDE P1PL1;
 %INCLUDE P2PL1;

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
 DCL INFILE FILE STREAM INPUT;
 DCL IN_BORO                    CHAR(01),
     IN_HOUSENUM                CHAR(12),
     IN_STREET_NAME             CHAR(32);

 /******** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT *********/
 DCL SYSPRINT FILE STREAM OUTPUT PRINT;
 ON ENDPAGE(SYSPRINT)
   PUT EDIT('SAMPLE PL1 PROGRAM #1 EXECUTION OUTPUT',
      '*****---------- INPUT ADDRESS ------------***** '||
      '*****---------------------'||
      ' SELECTED OUTPUT ITEMS -----------------------*****',
      'B HOUSE NUMBER IN-STREET-NAME                   '||
      '  ZIP CD NYPD-PCT SCHLDST '||
      'LOW CROSS STREET          HIGH CROSS STREET        ',
      '_ ____________ '||(32)'_'||' '||
      '_____ __ ________ _______ '||
      (25)'_'||' '||(25)'_')
```

PL/1 SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
     (PAGE,COL(1),A,SKIP(2),COL(1),A,COL(1),A,SKIP(0),COL(1),A);
 OPEN FILE(SYSPRINT) LINESIZE(133);
 SIGNAL ENDPAGE(SYSPRINT);
 /*****************************************************************/

 ON ENDFILE(INFILE)  BEGIN; EOF=YES; GOTO ENDLOOP;  END;
 OPEN FILE(INFILE);

 DO WHILE (EOF = NO);

   /********** REPLACE CODE BELOW WITH YOUR OWN INPUT *************/
   GET FILE(INFILE) EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME)
                        (COL(1),A(1),X(1),A(12),X(1),A(32));

   /*****************************************************************/
   /* TO MAKE A FUNCTION 1 CALL:                                    */
   /*   (1) INITIALIZE WORKAREA 1 TO SPACES                         */
   /*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                      */
   /*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD      */
   /*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER */
   /*       FIELD                                                   */
   /*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  */
   /*   (6) CALL GBI WITH 2 WORKAREAS                               */
   /*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS               */
   /*****************************************************************/
   WORK1PL1 = ' ';
   PIWA1_IN_FUNCTION_1 = '1';
   PIWA1_IN_BORO_1     = IN_BORO;
    /* for cow format the field house_number has length=16 */
   PIWA1_IN_HOUSENUM_DISPLAY  = IN_HOUSENUM;
   PIWA1_IN_STREET_1   = IN_STREET_NAME;
   PIWA1_IN_PLATFORM_INDICATOR = 'C';
   /***************************************************************/
   /***************************************************************/
   /* AS OF GEOSUPPORT 10.1,                                      */
   /* TO RECEIVE ROADBED-SPECIFIC INFORMATION,                    */
   /* SET THE ROADBED REQUEST SWITCH TO 'R', AS FOLLOWS:          */
   /* PIWA1_IN_ROADBED_REQ_SWITCH = 'R';                          */
   /***************************************************************/

   CALL GBI(P1PL1,P2PL1);

   IF PIWA1_OUT_RETURN_CODE ¼= '00' & PIWA1_OUT_RETURN_CODE ¼= '01'
   THEN DO;
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
          PUT EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME,
                   '*** FUNCTION 1 GRC =',PIWA1_OUT_RETURN_CODE,
                   'REASON CODE =',PIWA1_OUT_REASON_CODE,
                   '*** ',PIWA1_OUT_ERROR_MESSAGE)
                  (SKIP(2),COL(1),(7)(A,X(1)),SKIP(1),COL(49),A,A);
        END;
   ELSE DO;
          PUT EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME)
                  (SKIP(2),COL(1),(3)(A,X(1)));
          IF PIWA1_OUT_RETURN_CODE = '01'
          THEN DO;
                 /** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE **/
                 PUT EDIT('*** FUNCTION 1 WARNING, GRC =',
```

```
                 PIWA1_OUT_RETURN_CODE,
                 'REASON CODE =',PIWA1_OUT_REASON_CODE,
                 '*** ',PIWA1_OUT_ERROR_MESSAGE)
                (COL(49),(4)(A,X(1)),SKIP(1),COL(49),A,A);
      END;
     /*****************************************************/
     /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ****/
     /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS    ****/
     /*****************************************************/
     PUT EDIT(PIWA2_FN1_ZIP,PIWA2_FN1_COM_DIST_NUM,
              PIWA2_FN1_POL_PRECINCT,PIWA2_FN1_SCHL_DIST)
             (COL(49),(3)(A,X(1)),X(5),A);
     /***********************************************************/
     /* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND */
     /* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE   */
     /* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS    */
     /* FROM THE HIGH AND LOW STREET CODE LISTS CALL            */
     /* FUNCTION D:                                             */
     /*   (1) INITIALIZE WORKAREA 1 TO SPACES                   */
     /*   (2) SET WA1'S FUNCTION CODE FIELD TO D                */
     /*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED     */
     /*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE    */
     /*              HAS SPACE FOR ONLY 25 CHARACTERS)          */
     /*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN     */
     /*       STREET NAMES FORMATTED FOR DISPLAY                */
     /*   (5) MOVE WA2'S LOW B5SC FIELD TO WA1'S INPUT STREET   */
     /*       CODE 1 FIELD                                      */
     /*   (6) MOVE WA2'S HIGH B5SC FIELD TO WA1'S INPUT STREET  */
     /*       CODE 2 FIELD                                      */
     /*   (7) CALL GBI WITH 1 WORKAREA                          */
     /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS         */
     /***********************************************************/
     WORK1PL1 = ' ';
     PIWA1_IN_PLATFORM_INDICATOR = 'C';
     PIWA1_IN_FUNCTION_1        = 'D';
     PIWA1_IN_SNL               = '25';
     PIWA1_IN_SN_NORM_FORMAT    = 'C';

     PIWA1_IN_BORO_1  = SUBSTR(PIWA2_FN1_LOW_B5SC(1),1,1);
     PIWA1_IN_10SC_1  = SUBSTR(PIWA2_FN1_LOW_B5SC(1),2,5);
     PIWA1_IN_BORO_2  = SUBSTR(PIWA2_FN1_HI_B5SC(1),1,1);
     PIWA1_IN_10SC_2  = SUBSTR(PIWA2_FN1_HI_B5SC(1),2,5);


     CALL GBI(P1PL1);
     IF PIWA1_OUT_RETURN_CODE = '00'
     THEN DO;
            /******** INSERT YOUR OWN CODE HERE *******/
            PUT EDIT(PIWA1_OUT_STREET_1,PIWA1_OUT_STREET_2)
                    (COL(75),A(25),X(1),A(25));
      END;
     ELSE DO;
            /*** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE ***/
            PUT EDIT(IN_BORO,IN_HOUSENUM,IN_STREET_NAME,
                     '*** FUNCTION D GRC =',
                     PIWA1_OUT_RETURN_CODE,
                     'REASON CODE =',PIWA1_OUT_REASON_CODE,',',
                     '*** ',PIWA1_OUT_ERROR_MESSAGE)
```

PL/1 SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
                         (SKIP(2),COL(1),(8)(A,X(1)),
                          SKIP(1),COL(49),A,A);
               END;
        END;
 ENDLOOP:  END;
 CLOSE FILE(INFILE);
 END PL1C1SC;
/*
//LKED.SYSIN  DD *
  INCLUDE INCLIB(GBI)
/*
//INCLIB      DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                            *//
//*   THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP  *//
//*   MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:     *//
//*        A030.GEO.SUPPORT.PDSE.LOADLIB                        *//
//*        A030.GEO.SUPPORT.LOADLIB                             *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//GO.STEPLIB  DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//GO.SYSLMOD  DD DSN=&&GOSET,DISP=(OLD,DELETE)
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                            *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,  *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT     *//
//*   IS TAILORD TO USE STANDARD GEOSUPPORT DATA SET NAMES.     *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.   *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//GO.SYSPRINT DD SYSOUT=A
//GO.INFILE   DD *
1 22           READE ST
1 500          DUANE ST
1 2-4          BROADWAY
4 165-100      BAISLEY BLVD
4 165-1000     BAISLEY BLVD
/*
//
```

PL/1 SAMPLE PROGRAM #1 - Output Report

```
SAMPLE PL1 PROGRAM #1 EXECUTION OUTPUT

*****---------- INPUT ADDRESS ------------***** *****--------------------- SELECTED OUTPUT ITEMS -----------------------*****

B HOUSE NUMBER IN-STREET-NAME                    ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET         HIGH CROSS STREET
- ------------ -------------------------------- ----- -- -------- ------- ------------------------ ------------------------

1 22           READE ST                         10007 01 005      02      ELK STREET               BROADWAY

1 500          DUANE ST                         *** FUNCTION 1 GRC = 42 REASON CODE =
                                                *** ADDRESS NUMBER OUT OF RANGE

1 2-4          BROADWAY                         *** FUNCTION 1 WARNING, GRC = 01 REASON CODE = 1
                                                *** ADDR NUMBER ALTERED: RANGE ASSUMED. USING DIGITS BEFORE DASH ONLY
                                                10004 01 001      02      STONE STREET             BOWLING GREEN

4 165-100      BAISLEY BLVD                     11434 12 113      28      GUY R BREWER BOULEVARD   BEDELL STREET

4 165-1000     BAISLEY BLVD                     *** FUNCTION 1 GRC = 13 REASON CODE = 2
                                                *** ADDRESS NBR 165-1000  HAS MORE THAN 3 DIGITS AFTER THE DASH.
```

PL/1 SAMPLE PROGRAM #2

- **Input Job Stream - MSW**
- **Input Job Stream - COW**
- **Output Report**

PL/1 SAMPLE PROGRAM #2 - Job Stream - MSW

```
//PL1F2SRC  JOB YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  PL1 SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2. ******
//***                  (MSW FORMAT)                               ******
//*********************************************************************
//STEP1      EXEC IBMZCPLG,REGION=0M,GOPGM='PL1F2SR',
//           PARM.PLI='S,GS,INCLUDE',
//           PARM.LKED='AMODE(31)'
//PLI.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//SYSIN      DD *
 PL1F2SR:    PROC OPTIONS(MAIN);
   /*****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING  */
   /* TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.   */
   /* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION. */
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.   */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE          */
   /*       ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET        */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 2.            */
   /*****************************************************************/

 DCL EOF                      BIT(1)  INIT('0'B),
     YES                      BIT(1)  INIT('1'B),
     NO                       BIT(1)  INIT('0'B),
     ADDR                     BUILTIN,
     (I,J)                    FIXED BIN(15) INIT(0);

 /**************** GBI DECLARATION BELOW IS REQUIRED ****************/
 DCL GBI                      ENTRY OPTIONS(ASM,INTER);

 /******************************************************************/
 /** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BY THE %INCLUDE  **/
 /** STATEMENTS) IS STRONGLY ENCOURAGED                            **/
 /******************************************************************/
 %INCLUDE W1PL1;
 %INCLUDE W2PL1;

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
 DCL INFILE FILE STREAM INPUT;
 DCL IN_BORO1                 CHAR(01),
     IN_STREET_NAME1          CHAR(32),
     IN_BORO2                 CHAR(01),
     IN_STREET_NAME2          CHAR(32);

 /******** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT *********/
 DCL SYSPRINT FILE STREAM OUTPUT PRINT;
 ON ENDPAGE(SYSPRINT)
   PUT EDIT('SAMPLE PL1 PROGRAM #2 EXECUTION OUTPUT',
      '*****----------------- INPUT INTERSECTION '||(22)'-'||'***** '||
      '*****------------ SELECTED OUTPUT ITEMS -------------*****',
      'B IN-STREET-NAME-1'||(17)' '||'B IN-STREET-NAME-2'||(17)' '||
      '  ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES        ',
      '_ '||(32)'_'||' _ '||(32)'_'||' '||
      '_____ __ ________ _______ '||(32)'_')
     (PAGE,COL(1),A,SKIP(2),COL(1),A,COL(1),A,SKIP(0),COL(1),A);
```

PL/1 SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
OPEN FILE(SYSPRINT) LINESIZE(133);
SIGNAL ENDPAGE(SYSPRINT);
/*****************************************************************/

ON ENDFILE(INFILE)  BEGIN; EOF=YES; GOTO ENDLOOP;  END;
OPEN FILE(INFILE);

DO WHILE (EOF = NO);

  /********** REPLACE CODE BELOW WITH YOUR OWN INPUT *************/
  GET FILE(INFILE) EDIT(IN_BORO1,IN_STREET_NAME1,
                        IN_BORO2,IN_STREET_NAME2)
                       (COL(1),A(1),X(1),A(32),X(1),A(1),X(1),A(32));

  /*****************************************************************/
  /* TO MAKE A FUNCTION 2 CALL:                                    */
  /*   (1) INITIALIZE WORKAREA 1 TO SPACES                         */
  /*   (2) SET WA1'S FUNCTION-CODE TO 2                            */
  /*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   */
  /*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     */
  /*       FIELD                                                   */
  /*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD */
  /*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   */
  /*       FIELD                                                   */
  /*   (7) CALL GBI WITH 2 WORKAREAS                               */
  /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS               */
  /*****************************************************************/
  WORK1PL1 = ' ';
  GEO_WA1_IN_FUNCTION_1 = '2';
  GEO_WA1_IN_BORO       = IN_BORO1;
  GEO_WA1_IN_STREET_1   = IN_STREET_NAME1;
  GEO_WA1_IN_BORO_2     = IN_BORO2;
  GEO_WA1_IN_STREET_2   = IN_STREET_NAME2;

  CALL GBI(W1PL1,W2PL1);

  IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 ¼= '00' &
     GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 ¼= '01'
  THEN DO;
         /****** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *******/
         PUT EDIT('***** FUNCTION 2 GRC =',
                  GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2,
                  'REASON =',GEO_WA1_OUT_REASON_CODE,',',
                  GEO_WA1_OUT_ERROR_MESSAGE,
                  IN_BORO1,IN_STREET_NAME1,IN_BORO2,IN_STREET_NAME2)
                 (SKIP(2),COL(1),(3)(A,X(1)),A,A,X(1),A,
                  SKIP(1),(4)(A,X(1)));
       END;
  ELSE
      IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 = '01'
      THEN DO;
             /*** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE ****/
             PUT EDIT('***** FUNCTION 2 WARNING, GRC = '||
                      GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2||', '||
                      'REASON CODE = '||GEO_WA1_OUT_REASON_CODE||
                      ', '||GEO_WA1_OUT_ERROR_MESSAGE,
                      IN_BORO1,IN_STREET_NAME1,
                      IN_BORO2,IN_STREET_NAME2)
```

```
                    (SKIP(2),COL(1),A,SKIP(1),(4)(A,X(1)));
          END;

   IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 = '00'|
      GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 = '01'
   THEN DO;
         /*********************************************************/
         /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********/
         /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS.         *****/
         /*********************************************************/
         PUT EDIT(IN_BORO1,IN_STREET_NAME1,IN_BORO2,IN_STREET_NAME2,
                  GEO_WA2_FN2_ZIP,GEO_WA2_FN2_COMDIST_NUMBER,
                  GEO_WA2_FN2_POL_PRECINCT,GEO_WA2_FN2_SCHOOLDIST)
                 (SKIP(2),COL(1),(7)(A,X(1)),X(5),A);
         DO J = 1 TO GEO_WA2_FN2_NUM_OF_INTERSECTS;
           /*****************************************************/
           /* TO GET STREET NAMES FOR INTERSECTING STREET CODES   */
           /* MAKE A FUNCTION D CALL:                             */
           /*   (1) INITIALIZE WORKAREA 1 TO SPACES               */
           /*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D        */
           /*   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN */
           /*       STREET NAMES FORMATTED FOR DISPLAY            */
           /*   (4) MOVE THE PACKED BORO AND STREET CODE TO       */
           /*       WA1'S INPUT STREET CODE 1 FIELD               */
           /*   (5) CALL GBI WITH 1 WORKAREA                      */
           /*   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS     */
           /*****************************************************/
           WORK1PL1 = ' ';
           GEO_WA1_IN_FUNCTION_1        = 'D';
           GEO_WA1_IN_COMPACT_NAME_FLAG = 'C';
           GEO_WA1_IN_STREETCODE_1 = GEO_WA2_FN2_INTERSECT_PBSC(J);
           CALL GBI(W1PL1);
           IF GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2 = '00'
           THEN DO;
                  /******** INSERT YOUR OWN CODE HERE *******/
                  PUT EDIT(GEO_WA1_OUT_STREET_1)  (COL(97),A);
                END;
           ELSE DO;
                  /** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE **/
                  PUT EDIT('***** FUNCTION D GRC =',
                           GEO_WA1_OUT_RC_1||GEO_WA1_OUT_RC_2,
                           'REASON =',GEO_WA1_OUT_REASON_CODE,',',
                           GEO_WA1_OUT_ERROR_MESSAGE)
                          (SKIP(2),COL(1),(6)(A,X(1)));
                END;
         END;
       END;
 ENDLOOP:  END;
 CLOSE FILE(INFILE);
 END PL1F2SR;
/*
//LKED.SYSIN  DD *
  INCLUDE INCLIB(GBI)
/*
//INCLIB      DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                            *//
```

PL/1 SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
//*   THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP  *//
//*   MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:     *//
//*        A030.GEO.SUPPORT.PDSE.LOADLIB                        *//
//*        A030.GEO.SUPPORT.LOADLIB                             *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//GO.STEPLIB  DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//GO.SYSLMOD  DD DSN=&&GOSET,DISP=(OLD,DELETE)
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                            *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,  *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT     *//
//*   IS TAILORD TO USE STANDARD GEOSUPPORT DATA SET NAMES.     *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.   *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//GO.SYSPRINT DD SYSOUT=A
//GO.INFILE   DD *
1 CHAMBERS ST                     1 HUDSON ST
1 SIXTH AVE                       1 W. 8 ST
1 DUANE ST                        1 READE ST
/*
//
```

PL/1 SAMPLE PROGRAM #2 - Job Stream - COW

```
//PL1C2SRC  JOB YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  PL1 SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2. ******
//***                  (COW FORMAT)                             ******
//*********************************************************************
//STEP1      EXEC IBMZCPLG,REGION=0M,GOPGM='PL1C2SR',
//           PARM.PLI='S,GS,INCLUDE',
//           PARM.LKED='AMODE(31),LIST'
//PLI.SYSLIB DD DSN=A030.GEO.COPYLIB2,DISP=SHR
//           DD DSN=A030.GEO.COPYLIB,DISP=SHR
//SYSIN      DD *
 PL1C2SR:    PROC OPTIONS(MAIN);
   /*****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING  */
   /* TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.   */
   /* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION. */
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.   */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE          */
   /*       ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET        */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 2.            */
   /*****************************************************************/

 DCL EOF                       BIT(1)  INIT('0'B),
     YES                       BIT(1)  INIT('1'B),
     NO                        BIT(1)  INIT('0'B),
     ADDR                      BUILTIN,
     (I,J)                     FIXED BIN(15) INIT(0);

 /**************** GBI DECLARATION BELOW IS REQUIRED ****************/
 DCL GBI                       ENTRY OPTIONS(ASM,INTER);

 /*******************************************************************/
 /** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BY THE %INCLUDE  **/
 /** STATEMENTS) IS STRONGLY ENCOURAGED                            **/
 /*******************************************************************/
 %INCLUDE P1PL1;
 %INCLUDE P2PL1;

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
 DCL INFILE FILE STREAM INPUT;
 DCL IN_BORO1                  CHAR(01),
     IN_STREET_NAME1           CHAR(32),
     IN_BORO2                  CHAR(01),
     IN_STREET_NAME2           CHAR(32);

 /******** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT *********/
 DCL SYSPRINT FILE STREAM OUTPUT PRINT;
 ON ENDPAGE(SYSPRINT)
   PUT EDIT('SAMPLE PL1 PROGRAM #2 EXECUTION OUTPUT',
      '*****----------------- INPUT INTERSECTION '||(22)'-'||'***** '||
      '*****------------ SELECTED OUTPUT ITEMS -------------*****',
      'B IN-STREET-NAME-1'||(17)' '||'B IN-STREET-NAME-2'||(17)' '||
      '  ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES        ',
      '_ '||(32)'_'||' _ '||(32)'_'||' '||
      '_____ __ ________ _______ '||(32)'_')
     (PAGE,COL(1),A,SKIP(2),COL(1),A,COL(1),A,SKIP(0),COL(1),A);
```

PL/1 SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
OPEN FILE(SYSPRINT) LINESIZE(133);
SIGNAL ENDPAGE(SYSPRINT);
/*****************************************************************/

ON ENDFILE(INFILE)  BEGIN; EOF=YES; GOTO ENDLOOP;  END;
OPEN FILE(INFILE);

DO WHILE (EOF = NO);

  /********** REPLACE CODE BELOW WITH YOUR OWN INPUT *************/
  GET FILE(INFILE) EDIT(IN_BORO1,IN_STREET_NAME1,
                        IN_BORO2,IN_STREET_NAME2)
                       (COL(1),A(1),X(1),A(32),X(1),A(1),X(1),A(32));

  /*****************************************************************/
  /* TO MAKE A FUNCTION 2 CALL:                                    */
  /*   (1) INITIALIZE WORKAREA 1 TO SPACES                         */
  /*   (2) SET WA1'S FUNCTION-CODE TO 2                            */
  /*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   */
  /*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     */
  /*       FIELD                                                   */
  /*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD */
  /*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   */
  /*       FIELD                                                   */
  /*   (7) CALL GBI WITH 2 WORKAREAS                               */
  /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS               */
  /*****************************************************************/
  WORK1PL1 = ' ';
  PIWA1_IN_PLATFORM_INDICATOR = 'C';
  PIWA1_IN_FUNCTION_1 = '2';
  PIWA1_IN_BORO_1     = IN_BORO1;
  PIWA1_IN_STREET_1   = IN_STREET_NAME1;
  PIWA1_IN_BORO_2     = IN_BORO2;
  PIWA1_IN_STREET_2   = IN_STREET_NAME2;

  CALL GBI(P1PL1,P2PL1);

  IF PIWA1_OUT_RETURN_CODE ¼= '00' & PIWA1_OUT_RETURN_CODE ¼= '01'
  THEN DO;
         /****** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *******/
         PUT EDIT('***** FUNCTION 2 GRC =',PIWA1_OUT_RETURN_CODE,
                  'REASON =',PIWA1_OUT_REASON_CODE,',',
                  PIWA1_OUT_ERROR_MESSAGE,
                  IN_BORO1,IN_STREET_NAME1,IN_BORO2,IN_STREET_NAME2)
                 (SKIP(2),COL(1),(3)(A,X(1)),A,A,X(1),A,
                  SKIP(1),(4)(A,X(1)));
       END;
  ELSE
      IF PIWA1_OUT_RETURN_CODE = '01'
      THEN DO;
             /*** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE ****/
             PUT EDIT('***** FUNCTION 2 WARNING, GRC = '||
                      PIWA1_OUT_RETURN_CODE||', '||
                      'REASON CODE = '||PIWA1_OUT_REASON_CODE||
                      ', '||PIWA1_OUT_ERROR_MESSAGE,
                      IN_BORO1,IN_STREET_NAME1,
                      IN_BORO2,IN_STREET_NAME2)
                     (SKIP(2),COL(1),A,SKIP(1),(4)(A,X(1)));
```

PL/1 SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
            END;

    IF PIWA1_OUT_RETURN_CODE = '00'|
       PIWA1_OUT_RETURN_CODE = '01'
    THEN DO;
           /*************************************************************/
           /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********/
           /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS.         *****/
           /*************************************************************/
           PUT EDIT(IN_BORO1,IN_STREET_NAME1,IN_BORO2,IN_STREET_NAME2,
                    PIWA2_FN2_ZIP,PIWA2_FN2_COM_DIST_NUM,
                    PIWA2_FN2_POL_PRECINCT,PIWA2_FN2_SCHL_DIST)
                   (SKIP(2),COL(1),(7)(A,X(1)),X(5),A);
           DO J = 1 TO PIWA2_FN2_NUM_OF_INTERSECTS;
             /*********************************************************/
             /* TO GET STREET NAMES FOR INTERSECTING STREET CODES   */
             /* MAKE A FUNCTION D CALL:                             */
             /*   (1) INITIALIZE WORKAREA 1 TO SPACES               */
             /*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D        */
             /*   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN */
             /*       STREET NAMES FORMATTED FOR DISPLAY            */
             /*   (4) MOVE THE BORO AND STREET CODE TO WA1'S INPUT  */
             /*       STREET CODE 1 FIELD                           */
             /*   (5) CALL GBI WITH 1 WORKAREA                      */
             /*   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS     */
             /*********************************************************/
             WORK1PL1 = ' ';
             PIWA1_IN_PLATFORM_INDICATOR = 'C';
             PIWA1_IN_FUNCTION_1        = 'D';
             PIWA1_IN_SN_NORM_FORMAT    = 'C';
             PIWA1_IN_BORO_1  =
                     SUBSTR(PIWA2_FN2_INTERSECT_B5SC(1),1,1);
             PIWA1_IN_10SC_1 =
                     SUBSTR(PIWA2_FN2_INTERSECT_B5SC(J),2,5);
             CALL GBI(P1PL1);
             IF PIWA1_OUT_RETURN_CODE = '00'
             THEN DO;
                    /******** INSERT YOUR OWN CODE HERE *******/
                    PUT EDIT(PIWA1_OUT_STREET_1)  (COL(97),A);
                  END;
             ELSE DO;
                    /** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE **/
                    PUT EDIT('***** FUNCTION D GRC =',
                             PIWA1_OUT_RETURN_CODE,
                             'REASON =',PIWA1_OUT_REASON_CODE,',',
                             PIWA1_OUT_ERROR_MESSAGE)
                            (SKIP(2),COL(1),(6)(A,X(1)));
                  END;
           END;
         END;
  ENDLOOP:  END;
  CLOSE FILE(INFILE);
  END PL1C2SR;
/*
//LKED.SYSIN  DD *
  INCLUDE INCLIB(GBI)
/*
//INCLIB      DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
```

PL/1 SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                                 *//
//*   THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP       *//
//*   MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:          *//
//*        A030.GEO.SUPPORT.PDSE.LOADLIB                             *//
//*        A030.GEO.SUPPORT.LOADLIB                                  *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//GO.STEPLIB  DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//GO.SYSLMOD  DD DSN=&&GOSET,DISP=(OLD,DELETE)
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*   AS OF GEOSUPPORT VERSION 10.0,                                 *//
//*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,       *//
//*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT          *//
//*   IS TAILORD TO USE STANDARD GEOSUPPORT DATA SET NAMES.          *//
//*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.        *//
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*
//GO.SYSPRINT DD SYSOUT=A
//GO.INFILE   DD *
1 CHAMBERS ST                     1 HUDSON ST
1 SIXTH AVE                       1 W. 8 ST
1 DUANE ST                        1 READE ST
/*
//
```

PL/1 SAMPLE PROGRAM #2 - Output Report

```
SAMPLE PL1 PROGRAM #2 EXECUTION OUTPUT

*****---------------- INPUT INTERSECTION ----------------------***** *****------------ SELECTED OUTPUT ITEMS -------------*****

B IN-STREET-NAME-1                 B IN-STREET-NAME-2                   ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES
- -------------------------------- - -------------------------------- ----- -- -------- -------- --------------------------------

1 CHAMBERS ST                      1 HUDSON ST                        10007 01 001      02       CHAMBERS STREET
                                                                                                 HUDSON STREET
                                                                                                 WEST BROADWAY

1 SIXTH AVE                        1 W. 8 ST                          10014 02 006      02       6 AVENUE
                                                                                                 GREENWICH AVENUE
                                                                                                 WEST 8 STREET

***** FUNCTION 2 GRC = 62 REASON =  , READE STREET & DUANE STREET DO NOT INTERSECT
1 DUANE ST                         1 READE ST
```

C SAMPLE PROGRAM #1

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

C SAMPLE PROGRAM #1 - Job Stream -MSW

```
//CCCF1SRC JOB YOUR-JOB-CARD-INFORMATION
//*
//*******************************************************************
//***  C SAMPLE MSW BATCH GEOSUPPORT USER APPLICATION PROGRAM #1  ****
//*******************************************************************
//STEP1 EXEC EDCCLG,
//      CPARM='SS,OPT,OFFSET,SOURCE,XREF,LIST'
//COMPILE.SYSPRINT DD SYSOUT=A
//COMPILE.SYSLIB   DD
//                 DD
//                 DD  DSNAME=A030.GEO.COPYLIB,DISP=SHR
//COMPILE.SYSIN    DD *
   /****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING */
   /* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.*/
   /* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     */
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         */
   /*       ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET       */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.           */
   /****************************************************************/

#include <stdio.h>
#include <string.h>

 /*******************************************************************/
 /*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***/
 /*** #INCLUDE STATEMENTS) IS STRONGLY ENCOURAGED.                  ***/
 /*******************************************************************/
#include <wac.h>

 /**************   GBI OS LINKAGE BELOW IS REQUIRED   **************/
   #pragma linkage (GBI,OS)
   long GBI(void *, ...);

 /***** THE WORK AREA LAYOUTS MUST BE DECLARED USING THE TYPEDEFS ***/
 /***** IN THE GEOSUPPORT COPY FILE.                               ***/
   C_WA1 wa1;
   C_WA2_F1 wa2_f1;

void main ()
 {

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
    FILE *inpdat;
    struct tag
      {
       char in_boro;
       char filler1;
       char in_housenum         [12];
       char filler2;
       char in_street_name      [32];
       char filler3             [33];
      } recin ;

    inpdat = fopen("DD:INPDAT","rb");
    if (inpdat ==  NULL)
```

C SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
       {printf("INPDAT Data Set did not open.\n");
        return;}

 /***** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT  *****/
     printf("\fSAMPLE C PROGRAM #1 EXECUTION OUTPUT  ");
     printf("\n\n*****---------- INPUT ADDRESS ------------***** ***");
     printf("**--------------------- SELECTED OUTPUT ITEMS ---------");
     printf("--------------*****");
     printf("\n\nB HOUSE NUMBER IN-STREET-NAME              ");
     printf("       ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET ");
     printf("         HIGH CROSS STREET       ");
     printf("\r_ ____________ ____________________________");
     printf("____ _____ __ ________ _______ _________________");
     printf("________ _________________________");

  /*** REPLACE CODE BELOW WITH YOUR OWN INPUT ***/
  while (fread(&recin,1,sizeof(recin),inpdat))
   {

   /****************************************************************/
   /* TO MAKE A FUNCTION 1 CALL:                                   */
   /*   (1) INITIALIZE WORKAREA 1 TO SPACES                        */
   /*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                     */
   /*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD     */
   /*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER */
   /*       FIELD                                                  */
   /*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD */
   /*   (6) CALL GBI WITH 2 WORKAREAS                              */
   /*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS              */
   /****************************************************************/

    memset(&wa1,' ',sizeof(wa1));
    memcpy(wa1.input.func_code,"1 ",2);
    wa1.input.boro_1 = recin.in_boro ;
    memcpy(wa1.input.street_name_1,recin.in_street_name,32);
    memcpy(wa1.input.hse_nbr_disp,recin.in_housenum,12);
 /*
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
 /*     As of Geosupport Version 10.1,                              */
 /*     to receive roadbed-specific information,                    */
 /*     set the Roadbed Request Switch to 'R', as follows:          */
 /*     wa1.input.roadbedreq = 'R';                                 */
 /*                                                                 */
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
 /*                                                                 */
    GBI(&wa1,&wa2_f1);

    if ( (memcmp(wa1.output.ret_code,"01",2)) > 0
    ||   (memcmp(wa1.output.ret_code,"00",2)) < 0 )
         /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
     {
       printf("\n\n%c %.12s %.32s *** FUNCTION 1 GRC = %.2s"
              " REASON CODE = %c",
           recin.in_boro,recin.in_housenum,recin.in_street_name,
           wa1.output.ret_code,wa1.output.reject_reason_code) ;
       printf ("\n%51.5s %.80s", "***", wa1.output.msg) ;
     }
```

C SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
     if ( (memcmp(wa1.output.ret_code,"01",2)) == 0 )
          /******** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE *****/
      {
        printf("\n\n%c %.12s %.32s *** FUNCTION 1 WARNING, GRC = %.2s"
               " REASON CODE = %c",
            recin.in_boro,recin.in_housenum,recin.in_street_name,
            wa1.output.ret_code,wa1.output.reject_reason_code) ;
        printf ("\n%51.5s %.80s", "***", wa1.output.msg) ;
        printf("\n%47.1s %.5s %.2s %.3s      %.2s",
            " ",wa2_f1.zip_code,
            wa2_f1.com_dist+1,wa2_f1.police_pre,wa2_f1.com_schl_dist) ;
      }

     if ( (memcmp(wa1.output.ret_code,"00",2)) == 0 )
          /**********************************************************/
          /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********/
          /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS          *****/
          /**********************************************************/
        printf("\n\n%c %.12s %.31s  %.5s %.2s %.3s      %.2s",
            recin.in_boro,recin.in_housenum,recin.in_street_name,
            wa2_f1.zip_code,
            wa2_f1.com_dist+1,wa2_f1.police_pre,wa2_f1.com_schl_dist) ;

        /**********************************************************/
        /* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND  */
        /* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE    */
        /* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS     */
        /* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D: */
        /*   (1) INITIALIZE WORKAREA 1 TO SPACES                    */
        /*   (2) SET WA1'S FUNCTION CODE FIELD TO D                 */
        /*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED      */
        /*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE     */
        /*              HAS SPACE FOR ONLY 25 CHARACTERS)           */
        /*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN      */
        /*       STREET NAMES FORMATTED FOR DISPLAY                 */
        /*   (5) MOVE WA2'S LOW PBSC FIELD TO WA1'S INPUT STREET    */
        /*       CODE 1 FIELD                                       */
        /*   (6) MOVE WA2'S HIGH PBSC FIELD TO WA1'S INPUT STREET   */
        /*       CODE 2 FIELD                                       */
        /*   (7) CALL GBI WITH 1 WORKAREA                           */
        /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS          */
        /**********************************************************/
     if ( (memcmp(wa1.output.ret_code,"00",2)) == 0
     ||   (memcmp(wa1.output.ret_code,"01",2)) == 0 )
      {
        memset(&wa1,' ',sizeof(wa1));  /* Clear Work area 1 */
        wa1.input.func_code[0] = 'D' ;
        wa1.input.compact_flag = 'C' ;
        memcpy(wa1.input.snl,"25",2) ;
        memcpy(wa1.input.PB5SC_1,wa2_f1.l_x_sts[0],4) ;
        memcpy(wa1.input.PB5SC_2,wa2_f1.h_x_sts[0],4) ;
        GBI(&wa1);
        if ( (memcmp(wa1.output.ret_code,"00",2)) == 0 )
          /***** INSERT YOUR OWN CODE HERE ***********/
          printf("      %.25s %.25s",wa1.output.street_name_1,
                 wa1.output.street_name_2) ;
        else
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
```

C SAMPLE PROGRAM #1 - Job Stream - MSW (continued)

```
         {
          printf("\n\n%c %.12s %.32s *** FUNCTION D GRC = %.2s"
                 " REASON CODE = %c",
              recin.in_boro,recin.in_housenum,recin.in_street_name,
              wa1.output.ret_code,wa1.output.reject_reason_code) ;
          printf ("\n%51.5s %.80s", "***", wa1.output.msg) ;
         }

      }

   }

 }
/*
//LKED.SYSIN   DD *
    INCLUDE DD1(GBI)
/*
//LKED.DD1  DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, THE STEPLIB OR JOBLIB
//*  STATEMENTS OF THE GEOSUPPORT EXECUTION STEP MUST INCLUDE
//*  THE FOLLOWING TWO CONCATENATED DATA SETS IN THE SPECIFIED
//*  ORDER:
//*                A030.GEO.SUPPORT.PDSE.LOADLIB
//*                A030.GEO.SUPPORT.LOADLIB
//*
//*****************************************************************
//GO.STEPLIB  DD  DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, DD STATEMENTS FOR GEOSUPPORT
//*  DATA FILES (E.G. GRID, PAD, ETC) ARE NO LONGER NEEDED AND
//*  ARE IGNORED.  GEOSUPPORT IS TAILORED TO USE STANDARD
//*  GEOSUPPORT DATA SETS.  TO USE NON-STANDARD FILES, SEE YOUR
//*  SYSTEMS PROGRAMMER.
//*
//*****************************************************************
//GO.SYSPRINT DD SYSOUT=A
//GO.INPDAT DD *,DCB=LRECL=80
1 22          READE ST
1 500         DUANE ST
1 2-4         BROADWAY
4 165-100     BAISLEY BLVD
4 165-1000    BAISLEY BLVD
/*
//
```

C SAMPLE PROGRAM #1 - Job Stream - COW

```
//CCCC1SRC JOB YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  C SAMPLE COW BATCH GEOSUPPORT USER APPLICATION PROGRAM #1  ****
//*********************************************************************
//*
//STEP1 EXEC EDCCLG,
//      CPARM='SS,OPT,OFFSET,SOURCE,XREF,LIST'
//COMPILE.SYSPRINT DD SYSOUT=A
//COMPILE.SYSLIB   DD
//                 DD
//                 DD  DSNAME=A030.GEO.COPYLIB,DISP=SHR
//COMPILE.SYSIN    DD *
   /***************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING */
   /* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.*/
   /* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.     */
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         */
   /*       ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET       */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.           */
   /***************************************************************/

#include <stdio.h>
#include <string.h>

 /*******************************************************************/
 /*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***/
 /*** #INCLUDE STATEMENTS) IS STRONGLY ENCOURAGED.                  ***/
 /*******************************************************************/
#include <pac.h>

 /**************   GBI OS LINKAGE BELOW IS REQUIRED   **************/
#pragma linkage (GBI,OS)
long GBI(void *, ...);

 /***** THE WORK AREA LAYOUTS MUST BE DECLARED USING THE TYPEDEFS ***/
 /***** IN THE GEOSUPPORT COPY FILE.                               ***/
   C_WA1 wa1;
   C_WA2_F1 wa2_f1;

void main ()
 {

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
    FILE *inpdat;
    struct tag
      {
       char in_boro;
       char filler1;
       char in_housenum         [12];
       char filler2;
       char in_street_name      [32];
       char filler3             [33];
      } recin ;

    inpdat = fopen("DD:INPDAT","rb");
```

C SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
    if (inpdat ==  NULL)
        {printf("INPDAT Data Set did not open.\n");
         return;}

 /***** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT  *****/
     printf("\fSAMPLE C PROGRAM #1 EXECUTION OUTPUT  ");
     printf("\n\n*****---------- INPUT ADDRESS -----------***** ***");
     printf("**-------------------- SELECTED OUTPUT ITEMS ---------");
     printf("-------------*****");
     printf("\n\nB HOUSE NUMBER IN-STREET-NAME              ");
     printf("       ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET ");
     printf("         HIGH CROSS STREET       ");
     printf("\r_ ____________ ____________________________");
     printf("____ _____ __ ________ _______ _________________");
     printf("________ _________________________");

  /*** REPLACE CODE BELOW WITH YOUR OWN INPUT ***/
  while (fread(&recin,1,sizeof(recin),inpdat))
   {

   /******************************************************************/
   /* TO MAKE A FUNCTION 1 CALL:                                     */
   /*   (1) INITIALIZE WORKAREA 1 TO SPACES                          */
   /*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                       */
   /*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD       */
   /*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER */
   /*       FIELD                                                    */
   /*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD  */
   /*   (6) CALL GBI WITH 2 WORKAREAS                                */
   /*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS                */
   /******************************************************************/

    memset(&wa1,' ',sizeof(wa1));
    memcpy(wa1.input.func_code,"1 ",2);
    wa1.input.sti??(0??).boro = recin.in_boro ;
    memcpy(wa1.input.sti??(0??).Street_name,recin.in_street_name,32);
           /* Please note that the house number field is actually */
           /* 16 bytes.  If you are only using 12 bytes, it is    */
           /* critical that you blank out the work area before    */
           /* you move in the house number                        */
    memcpy(wa1.input.hse_nbr_disp,recin.in_housenum,12);
    wa1.input.platform_ind = 'C';    /* Tells Geosupport that you */
                                     /* are using the character   */
 /*
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
 /*     As of Geosupport Version 10.1,                             */
 /*     to receive roadbed-specific information,                   */
 /*     set the Roadbed Request Switch to 'R', as follows:         */
 /*     wa1.input.roadbedrequest = 'R';                            */
 /*                                                                */
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
 /*  */


    GBI(&wa1,&wa2_f1);

    if ( (memcmp(wa1.output.ret_code,"01",2)) > 0
    ||   (memcmp(wa1.output.ret_code,"00",2)) < 0 )
```

C SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
     /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
  {
    printf("\n\n%c %.12s %.32s *** FUNCTION 1 GRC = %.2s"
           " REASON CODE = %c",
        recin.in_boro,recin.in_housenum,recin.in_street_name,
        wa1.output.ret_code,wa1.output.reason_code) ;
    printf ("\n%51.5s %.80s", "***", wa1.output.msg) ;
  }

 if ( (memcmp(wa1.output.ret_code,"01",2)) == 0 )
     /******** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE *****/
  {
    printf("\n\n%c %.12s %.32s *** FUNCTION 1 WARNING, GRC = %.2s"
           " REASON CODE = %c",
        recin.in_boro,recin.in_housenum,recin.in_street_name,
        wa1.output.ret_code,wa1.output.reason_code) ;
    printf ("\n%51.5s %.80s", "***", wa1.output.msg) ;
    printf("\n%47.1s %.5s %.2s %.3s      %.2s",
        " ",wa2_f1.zip_code,
        wa2_f1.com_dist+1,wa2_f1.police_pre,wa2_f1.com_schl_dist) ;
  }

 if ( (memcmp(wa1.output.ret_code,"00",2)) == 0 )
     /*********************************************************/
     /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********/
     /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS          *****/
     /*********************************************************/
    printf("\n\n%c %.12s %.31s  %.5s %.2s %.3s      %.2s",
        recin.in_boro,recin.in_housenum,recin.in_street_name,
        wa2_f1.zip_code,
        wa2_f1.com_dist+1,wa2_f1.police_pre,wa2_f1.com_schl_dist) ;

    /***********************************************************/
    /* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND  */
    /* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE    */
    /* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS     */
    /* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D: */
    /*   (1) INITIALIZE WORKAREA 1 TO SPACES                    */
    /*   (2) SET WA1'S FUNCTION CODE FIELD TO D                 */
    /*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED      */
    /*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE     */
    /*              HAS SPACE FOR ONLY 25 CHARACTERS)           */
    /*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN      */
    /*       STREET NAMES FORMATTED FOR DISPLAY                 */
    /*   (5) MOVE WA2'S LOW PBSC FIELD TO WA1'S INPUT STREET    */
    /*       CODE 1 FIELD                                       */
    /*   (6) MOVE WA2'S HIGH PBSC FIELD TO WA1'S INPUT STREET   */
    /*       CODE 2 FIELD                                       */
    /*   (7) CALL GBI WITH 1 WORKAREA                           */
    /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS          */
    /***********************************************************/
 if ( (memcmp(wa1.output.ret_code,"00",2)) == 0
 ||   (memcmp(wa1.output.ret_code,"01",2)) == 0 )
  {
    memset(&wa1,' ',sizeof(wa1));  /* Clear Work area 1 */
    wa1.input.func_code[0] = 'D' ;
    wa1.input.st_name_norm = 'C' ;
    memcpy(wa1.input.snl,"25",2) ;
```

```
        wa1.input.platform_ind = 'C';
        wa1.input.sti??(0??).boro=wa2_f1.st??(0??).B5SC??(0??)??(0??);
        memcpy(wa1.input.sti??(0??).SC10,
               wa2_f1.st??(0??).B5SC??(0??)+1,5);
        wa1.input.sti??(1??).boro=wa2_f1.st??(1??).B5SC??(0??)??(0??);
        memcpy(wa1.input.sti??(1??).SC10,
               wa2_f1.st??(1??).B5SC??(0??)+1,5);
        GBI(&wa1);
        if ( (memcmp(wa1.output.ret_code,"00",2)) == 0 )
          /***** INSERT YOUR OWN CODE HERE ***********/
          printf("      %.25s %.25s",wa1.output.sto??(0??).Street_name,
                 wa1.output.sto??(1??).Street_name) ;
        else
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
         {
          printf("\n\n%c %.12s %.32s *** FUNCTION D GRC = %.2s"
                 " REASON CODE = %c",
              recin.in_boro,recin.in_housenum,recin.in_street_name,
              wa1.output.ret_code,wa1.output.reason_code) ;
          printf ("\n%51.5s %.80s", "***", wa1.output.msg) ;
         }

      }

   }

 }
/*
//LKED.SYSIN   DD *
    INCLUDE DD1(GBI)
/*
//LKED.DD1  DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, THE STEPLIB OR JOBLIB
//*  STATEMENTS OF THE GEOSUPPORT EXECUTION STEP MUST INCLUDE
//*  THE FOLLOWING TWO CONCATENATED DATA SETS IN THE SPECIFIED
//*  ORDER:
//*                 A030.GEO.SUPPORT.PDSE.LOADLIB
//*                 A030.GEO.SUPPORT.LOADLIB
//*
//*****************************************************************
//GO.STEPLIB  DD  DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, DD STATEMENTS FOR GEOSUPPORT
//*  DATA FILES (E.G. GRID, PAD, ETC) ARE NO LONGER NEEDED AND
//*  ARE IGNORED.  GEOSUPPORT IS TAILORED TO USE STANDARD
//*  GEOSUPPORT DATA SETS.  TO USE NON-STANDARD FILES, SEE YOUR
//*  SYSTEMS PROGRAMMER.
//*
//*****************************************************************
//GO.SYSPRINT DD SYSOUT=A
//GO.INPDAT DD *,DCB=LRECL=80
1 22          READE ST
1 500         DUANE ST
1 2-4         BROADWAY
```

C SAMPLE PROGRAM #1 - Job Stream - COW (continued)

```
4 165-100      BAISLEY BLVD
4 165-1000     BAISLEY BLVD
/*
//
```

C SAMPLE PROGRAM #1 - Output Report

```
SAMPLE C PROGRAM #1 EXECUTION OUTPUT

*****---------- INPUT ADDRESS -----------***** *****-------------------- SELECTED OUTPUT ITEMS -----------------------*****

B HOUSE NUMBER IN-STREET-NAME                     ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET          HIGH CROSS STREET
- ------------ -------------------------------- ----- -- -------- ------- ------------------------ ------------------------

1 22           READE ST                         10007 01 005      02      ELK STREET                BROADWAY

1 500          DUANE ST                         *** FUNCTION 1 GRC = 42 REASON CODE =
                                                *** ADDRESS NUMBER OUT OF RANGE

1 2-4          BROADWAY                         *** FUNCTION 1 WARNING, GRC = 01 REASON CODE = 1
                                                *** ADDR NUMBER ALTERED: RANGE ASSUMED. USING DIGITS BEFORE DASH ONLY
                                                10004 01 001      02      STONE STREET              BOWLING GREEN

4 165-100      BAISLEY BLVD                     11434 12 113      28      GUY R BREWER BOULEVARD    BEDELL STREET

4 165-1000     BAISLEY BLVD                     *** FUNCTION 1 GRC = 13 REASON CODE = 2
                                                *** ADDRESS NBR 165-1000  HAS MORE THAN 3 DIGITS AFTER THE DASH.
```

C SAMPLE PROGRAM #2

- Input Job Stream - MSW

- Input Job Stream - COW

- Output Report

C SAMPLE PROGRAM #2 - Job Stream - MSW

```
//CCCF2SRC JOB YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  C SAMPLE MSW BATCH GEOSUPPORT USER APPLICATION PROGRAM #2  ****
//*********************************************************************
//*
//STEP1 EXEC EDCCLG,
//      CPARM='SS,OPT,OFFSET,SOURCE,XREF,LIST'
//COMPILE.SYSPRINT DD SYSOUT=A
//COMPILE.SYSLIB   DD
//                 DD
//                 DD  DSNAME=A030.GEO.COPYLIB,DISP=SHR
//COMPILE.SYSIN    DD *
   /****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING */
   /* TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.  */
   /* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION.*/
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         */
   /*       ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET       */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 2.           */
   /****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

 /*******************************************************************/
 /*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***/
 /*** #INCLUDE STATEMENTS) IS STRONGLY ENCOURAGED.                  ***/
 /*******************************************************************/
#include <wac.h>

 /**************   GBI DECLARATION BELOW IS REQUIRED   **************/
   #pragma linkage (GBI,OS)
   long GBI(void *, ...);

 /***** THE WORK AREA LAYOUTS MUST BE DECLARED USING THE TYPEDEFS ***/
 /***** IN THE GEOSUPPORT COPY FILE.                              ***/
   C_WA1 wa1;
   C_WA2_F2 wa2_f2;

void main ()
 {

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
    FILE *infile;
    struct tag
      {
       char in_boro1;
       char filler1;
       char in_street_name1     [32];
       char filler2;
       char in_boro2;
       char filler3;
       char in_street_name2     [32];
       char filler4             [11];
```

C SAMPLE PROGRAM #2 - Job Stream - MSW (continued)

```
      } recin ;
   short int j, i;
   char temp [2];
    if ((infile = fopen("DD:INFILE","rb")) ==  NULL)
        {printf("INFILE Data Set did not open.\n");
         return;}

 /***** REPLACE CODE BELOW WITH YOUR OWN REPRORT LAYOUT  *****/
      printf("\fSAMPLE C PROGRAM #2 EXECUTION OUTPUT  ");
      printf("\n\n*****---------------- INPUT INTERSECTION ---------");
      printf("------------***** *****----------- SELECTED OUTPUT ");
      printf("ITEMS ------------*****");
      printf("\n\nB IN-STREET-NAME-1            ");
      printf("     B IN-STREET-NAME-2                   ZIP CD");
      printf(" NYPD-PCT SCHLDST INTERSECTING STREET NAMES");
      printf(  "\r_ ____________________________");
      printf("____ _ ________________________________ _____ __");
      printf(" ________ _______ _________________________________");

  /*** REPLACE CODE BELOW WITH YOUR OWN INPUT ***/
  while (fread(&recin,1,sizeof(recin),infile))
   {

   /***************************************************************/
   /* TO MAKE A FUNCTION 2 CALL:                                  */
   /*   (1) INITIALIZE WORKAREA 1 TO SPACES                       */
   /*   (2) SET WA1'S FUNCTION-CODE TO 2                          */
   /*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD   */
   /*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME     */
   /*       FIELD                                                 */
   /*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD */
   /*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2   */
   /*       FIELD                                                 */
   /*   (7) CALL GBI WITH 2 WORKAREAS                             */
   /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS             */
   /***************************************************************/

     memset(&wa1,' ',sizeof(wa1));
     memcpy(wa1.input.func_code,"2 ",2);
     wa1.input.boro_1 = recin.in_boro1 ;
     memcpy(wa1.input.street_name_1,recin.in_street_name1,32);
     wa1.input.boro_2 = recin.in_boro2 ;
     memcpy(wa1.input.street_name_2,recin.in_street_name2,32);
     GBI(&wa1,&wa2_f2);

     if ( (memcmp(wa1.output.ret_code,"01",2)) > 0
     ||   (memcmp(wa1.output.ret_code,"00",2)) < 0 )
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
      {
        printf("\n\n***** FUNCTION 2 GRC = %.2s"
               " REASON CODE = %c. %.80s",
            wa1.output.ret_code,wa1.output.reject_reason_code,
            wa1.output.msg) ;
        printf
        ("\n%c %.32s %c %.32s ",
            recin.in_boro1,recin.in_street_name1,recin.in_boro2,
            recin.in_street_name2) ;
      }
```

```
     if ( (memcmp(wa1.output.ret_code,"01",2)) == 0 )
          /******** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE *****/
      {
        printf("\n\n***** FUNCTION 2 WARNING GRC = %.2s"
               " REASON CODE = %c. %.80s",
            wa1.output.ret_code,wa1.output.reject_reason_code,
            wa1.output.msg) ;
        printf
        ("\n%c %.32s %c %.32s ",
            recin.in_boro1,recin.in_street_name1,recin.in_boro2,
            recin.in_street_name2) ;
      }

     if ( (memcmp(wa1.output.ret_code,"00",2)) == 0
     ||   (memcmp(wa1.output.ret_code,"01",2)) == 0 )
          /**********************************************************/
          /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********/
          /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS          *****/
          /**********************************************************/
      {
       printf("\n\n%c %.32s %c %.32s %.5s %.2s %.3s      %.2s     ",
            recin.in_boro1,recin.in_street_name1,recin.in_boro2,
            recin.in_street_name2,wa2_f2.zip_code,
            wa2_f2.com_dist+1,wa2_f2.police_pre,wa2_f2.com_schl_dist) ;

       temp [0] = wa2_f2.nbr_x_sts ;
       temp [1] = 0;
       i = atoi(temp) ;
       for (j=0; j<i; j++)
       {
            /******************************************************/
            /* TO GET STREET NAMES FOR INTERSECTING STREET CODES   */
            /* MAKE A FUNCTION D CALL:                             */
            /*   (1) INITIALIZE WORKAREA 1 TO SPACES               */
            /*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D        */
            /*   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN */
            /*       STREET NAMES FORMATTED FOR DISPLAY            */
            /*   (4) MOVE THE PACKED BORO AND STREET CODE TO       */
            /*       WA1'S INPUT STREET CODE 1 FIELD               */
            /*   (5) CALL GBI WITH 1 WORKAREA                      */
            /*   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS     */
            /******************************************************/
        memset(&wa1,' ',sizeof(wa1));
        wa1.input.func_code[0] = 'D' ;
        wa1.input.compact_flag = 'C' ;
        memcpy(wa1.input.PB5SC_1,wa2_f2.x_sts[j],4) ;
        GBI(&wa1);
        if ( (memcmp(wa1.output.ret_code,"00",2)) == 0 )
         {
          /***** INSERT YOUR OWN CODE HERE ***********/
          if (j==0)
            printf(" %.32s",wa1.output.street_name_1);
          else
            printf("\n%128.32s",wa1.output.street_name_1);
         }
        else
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
          printf("\n\n***** FUNCTION D GRC = %.2s"
```

```
              " REASON CODE = %c. %.80s",
           wa1.output.ret_code,wa1.output.reject_reason_code,
           wa1.output.msg) ;
       }

      }

   }

 }
/*
//LKED.SYSIN   DD *
    INCLUDE DD1(GBI)
/*
//LKED.DD1  DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, THE STEPLIB OR JOBLIB
//*  STATEMENTS OF THE GEOSUPPORT EXECUTION STEP MUST INCLUDE
//*  THE FOLLOWING TWO CONCATENATED DATA SETS IN THE SPECIFIED
//*  ORDER:
//*                A030.GEO.SUPPORT.PDSE.LOADLIB
//*                A030.GEO.SUPPORT.LOADLIB
//*
//*****************************************************************
//GO.STEPLIB  DD  DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//*****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, DD STATEMENTS FOR GEOSUPPORT
//*  DATA FILES (E.G. GRID, PAD, ETC) ARE NO LONGER NEEDED AND
//*  ARE IGNORED.  GEOSUPPORT IS TAILORED TO USE STANDARD
//*  GEOSUPPORT DATA SETS.  TO USE NON-STANDARD FILES, SEE YOUR
//*  SYSTEMS PROGRAMMER.
//*
//*****************************************************************
//GO.SYSPRINT DD SYSOUT=A
//GO.INFILE DD *
1 CHAMBERS ST                      1 HUDSON ST
1 SIXTH AVE                        1 W. 8 ST
1 DUANE ST                         1 READE ST
/*
//
```

C SAMPLE PROGRAM #2 - Job Stream - COW

```
//CCCC2SRC JOB   YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  C SAMPLE COW BATCH GEOSUPPORT USER APPLICATION PROGRAM #2  ****
//*********************************************************************
//*
//STEP1 EXEC EDCCLG,
//      CPARM='SS,OPT,OFFSET,SOURCE,XREF,LIST'
//COMPILE.SYSPRINT DD SYSOUT=A
//COMPILE.SYSLIB   DD
//                 DD
//                 DD  DSNAME=A030.GEO.COPYLIB,DISP=SHR
//COMPILE.SYSIN    DD *
   /*****************************************************************/
   /* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING */
   /* TWO BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.  */
   /* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION.*/
   /* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.  */
   /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   /* NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE         */
   /*       ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET       */
   /*       NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 2.           */
   /*****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

 /********************************************************************/
 /*** USE OF GEOSUPPORT COPY LIBRARIES (REFERENCED BELOW BY THE     ***/
 /*** #INCLUDE STATEMENTS) IS STRONGLY ENCOURAGED.                  ***/
 /********************************************************************/
#include <pac.h>

 /**************   GBI DECLARATION BELOW IS REQUIRED   **************/
   #pragma linkage (GBI,OS)
   long GBI(void *, ...);

 /***** THE WORK AREA LAYOUTS MUST BE DECLARED USING THE TYPEDEFS ***/
 /***** IN THE GEOSUPPORT COPY FILE.                               ***/
   C_WA1 wa1;
   C_WA2_F2 wa2_f2;

void main ()
 {

 /***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION *****/
    FILE *infile;
    struct tag
      {
       char in_boro1;
       char filler1;
       char in_street_name1     [32];
       char filler2;
       char in_boro2;
       char filler3;
       char in_street_name2     [32];
       char filler4             [11];
```

C SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
      } recin ;
   short int j, i;
   char temp [2];
    if ((infile = fopen("DD:INFILE","rb")) ==  NULL)
        {printf("INFILE Data Set did not open.\n");
         return;}

 /***** REPLACE CODE BELOW WITH YOUR OWN REPRORT LAYOUT  *****/
      printf("\fSAMPLE C PROGRAM #2 EXECUTION OUTPUT  ");
      printf("\n\n*****---------------- INPUT INTERSECTION ---------");
      printf("------------***** *****----------- SELECTED OUTPUT ");
      printf("ITEMS ------------*****");
      printf("\n\nB IN-STREET-NAME-1            ");
      printf("     B IN-STREET-NAME-2                  ZIP CD");
      printf(" NYPD-PCT SCHLDST INTERSECTING STREET NAMES");
      printf(  "\r_ ____________________________");
      printf("____ _ ________________________________ _____ __");
      printf(" ________ _______ _________________________________");

  /*** REPLACE CODE BELOW WITH YOUR OWN INPUT ***/
  while (fread(&recin,1,sizeof(recin),infile))
   {

   /*****************************************************************/
   /* TO MAKE A FUNCTION 2 CALL:                                    */
   /*   (1) INITIALIZE WORKAREA 1 TO SPACES                         */
   /*   (2) SET WA1'S FUNCTION-CODE TO 2                            */
   /*   (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD  */
   /*   (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME    */
   /*       FIELD                                                   */
   /*   (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD */
   /*   (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2  */
   /*       FIELD                                                   */
   /*   (7) CALL GBI WITH 2 WORKAREAS                               */
   /*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS               */
   /*****************************************************************/

     memset(&wa1,' ',sizeof(wa1));
     memcpy(wa1.input.func_code,"2 ",2);
     wa1.input.sti??(0??).boro = recin.in_boro1 ;
     memcpy(wa1.input.sti??(0??).Street_name,recin.in_street_name1,32);
     wa1.input.sti??(1??).boro = recin.in_boro2 ;
     memcpy(wa1.input.sti??(1??).Street_name,recin.in_street_name2,32);
     wa1.input.platform_ind = 'C';    /* Tells Geosupport that you */
                                      /* are using the character   */
                                      /* only work areas           */
     GBI(&wa1,&wa2_f2);

     if ( (memcmp(wa1.output.ret_code,"01",2)) > 0
     ||   (memcmp(wa1.output.ret_code,"00",2)) < 0 )
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
      {
        printf("\n\n***** FUNCTION 2 GRC = %.2s"
               " REASON CODE = %c. %.80s",
            wa1.output.ret_code,wa1.output.reason_code,
            wa1.output.msg) ;
        printf
        ("\n%c %.32s %c %.32s ",
```

```
           recin.in_boro1,recin.in_street_name1,recin.in_boro2,
           recin.in_street_name2) ;
    }

   if ( (memcmp(wa1.output.ret_code,"01",2)) == 0 )
        /******** INSERT YOUR OWN WARNING HANDLING ROUTINE HERE *****/
    {
      printf("\n\n***** FUNCTION 2 WARNING GRC = %.2s"
             " REASON CODE = %c. %.80s",
           wa1.output.ret_code,wa1.output.reason_code,
           wa1.output.msg) ;
      printf
      ("\n%c %.32s %c %.32s ",
           recin.in_boro1,recin.in_street_name1,recin.in_boro2,
           recin.in_street_name2) ;
    }

   if ( (memcmp(wa1.output.ret_code,"00",2)) == 0
   ||   (memcmp(wa1.output.ret_code,"01",2)) == 0 )
        /***********************************************************/
        /***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR ***********/
        /***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS          *****/
        /***********************************************************/
    {
     printf("\n\n%c %.32s %c %.32s %.5s %.2s %.3s      %.2s     ",
           recin.in_boro1,recin.in_street_name1,recin.in_boro2,
           recin.in_street_name2,wa2_f2.zip_code,
           wa2_f2.com_dist+1,wa2_f2.police_pre,wa2_f2.com_schl_dist) ;

     temp [0] = wa2_f2.inter.nbr_sts ;
     temp [1] = 0;
     i = atoi(temp) ;
     for (j=0; j<i; j++)
     {
           /*******************************************************/
           /* TO GET STREET NAMES FOR INTERSECTING STREET CODES   */
           /* MAKE A FUNCTION D CALL:                             */
           /*   (1) INITIALIZE WORKAREA 1 TO SPACES               */
           /*   (2) SET THE WA1'S FUNCTION CODE FIELD TO D        */
           /*   (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN */
           /*       STREET NAMES FORMATTED FOR DISPLAY            */
           /*   (4) MOVE THE PACKED BORO AND STREET CODE TO       */
           /*       WA1'S INPUT STREET CODE 1 FIELD               */
           /*   (5) CALL GBI WITH 1 WORKAREA                      */
           /*   (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS     */
           /*******************************************************/
      memset(&wa1,' ',sizeof(wa1));
      wa1.input.func_code[0] = 'D' ;
      wa1.input.st_name_norm = 'C' ;
      wa1.input.platform_ind = 'C';
      wa1.input.sti??(0??).boro = wa2_f2.inter.B5SC??(j??)??(0??);
      memcpy(wa1.input.sti??(0??).SC10,wa2_f2.inter.B5SC??(j??)+1,5);
      GBI(&wa1);
      if ( (memcmp(wa1.output.ret_code,"00",2)) == 0 )
       {
        /***** INSERT YOUR OWN CODE HERE ***********/
        if (j==0)
          printf(" %.32s",wa1.output.sto??(0??).Street_name);
```

C SAMPLE PROGRAM #2 - Job Stream - COW (continued)

```
          else
            printf("\n%128.32s",wa1.output.sto??(0??).Street_name);
           }
         else
          /******** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE *****/
          printf("\n\n***** FUNCTION D GRC = %.2s"
               " REASON CODE = %c. %.80s",
            wa1.output.ret_code,wa1.output.reason_code,
            wa1.output.msg) ;
        }

       }

    }

 }
/*
//LKED.SYSIN   DD *
    INCLUDE DD1(GBI)
/*
//LKED.DD1  DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, THE STEPLIB OR JOBLIB
//*  STATEMENTS OF THE GEOSUPPORT EXECUTION STEP MUST INCLUDE
//*  THE FOLLOWING TWO CONCATENATED DATA SETS IN THE SPECIFIED
//*  ORDER:
//*                A030.GEO.SUPPORT.PDSE.LOADLIB
//*                A030.GEO.SUPPORT.LOADLIB
//*
//****************************************************************
//GO.STEPLIB  DD  DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD  DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//****************************************************************
//*
//*  AS OF GEOSUPPORT VERSION 10.0, DD STATEMENTS FOR GEOSUPPORT
//*  DATA FILES (E.G. GRID, PAD, ETC) ARE NO LONGER NEEDED AND
//*  ARE IGNORED.  GEOSUPPORT IS TAILORED TO USE STANDARD
//*  GEOSUPPORT DATA SETS.  TO USE NON-STANDARD FILES, SEE YOUR
//*  SYSTEMS PROGRAMMER.
//*
//****************************************************************
//GO.SYSPRINT DD SYSOUT=A
//GO.INFILE DD *
1 CHAMBERS ST                    1 HUDSON ST
1 SIXTH AVE                      1 W. 8 ST
1 DUANE ST                       1 READE ST
/*
//
```

C SAMPLE PROGRAM #2 - Output Report

```
SAMPLE C PROGRAM #2 EXECUTION OUTPUT

*****----------------- INPUT INTERSECTION ----------------------***** *****------------ SELECTED OUTPUT ITEMS -------------*****

B IN-STREET-NAME-1                 B IN-STREET-NAME-2                   ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES
- -------------------------------- - -------------------------------- ----- -- -------- ------- --------------------------------

1 CHAMBERS ST                      1 HUDSON ST                        10007 01 001      02      CHAMBERS STREET
                                                                                                HUDSON STREET
                                                                                                WEST BROADWAY

1 SIXTH AVE                        1 W. 8 ST                          10014 02 006      02      6 AVENUE
                                                                                                GREENWICH AVENUE
                                                                                                WEST 8 STREET

***** FUNCTION 2 GRC = 62 REASON CODE =  . READE STREET & DUANE STREET DO NOT INTERSECT
1 DUANE ST                         1 READE ST
```

NATURAL SAMPLE PROGRAM #1

- Program Source Code - MSW

- Program Source Code - COW

- Input Job Stream

- Output Report

NATURAL SAMPLE PROGRAM #1 - Program Source Code - MSW

```
************************************************************************
* PGM NAME: GEOBUPG1    DATE: 08-18-98                                 *
*                                                                      *
* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING        *
* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.       *
* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.            *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  *
*         NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE        *
*               ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET      *
*               NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.          *
************************************************************************
*                                                                      *
************************************************************************
* USE OF GEOSUPPORT LDA (REFERENCED BELOW BY THE LOCAL USING STATEMENT)*
* IS STRONGLY ENCOURAGED.                                              *
************************************************************************
DEFINE DATA
 LOCAL USING GEOLW1
 LOCAL USING GEOLW2
*
***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION      *****
 LOCAL
  01 #USER-INPUT
     02 #USER-BORO                   (A1)
     02 #FILLER1                     (A1)
     02 #USER-HSE-NUM                (A12)
     02 #FILLER2                     (A1)
     02 #USER-STRT-NAME              (A32)
     02 #FILLER3                     (A33)
*
  01 #SAVE-RET-CODE                  (A2)
  01 #OUT-STREET-1-SNL25             (A25)
  01 #OUT-STREET-2-SNL25             (A25)
  01 #OUT-ERROR-MESSAGE-77           (A77)
*
END-DEFINE
*
FORMAT LS=133 PS=65
*
***** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT               *****
WRITE NOTITLE
 1T'SAMPLE NATURAL PROGRAM #1 EXECUTION OUTPUT'//
 1T'*****---------- INPUT ADDRESS -----------*****'
 49T'*****--------------------- SELECTED OUTPUT'
 92T'ITEMS ----------------------*****'//
 1T'B HOUSE NUMBER IN-STREET-NAME                     ZIP CD'
 58T'NYPD-PCT SCHLDST LOW CROSS STREET         '
 101T'HIGH CROSS STREET        '/
 1T'- ------------ -------------------------------- ----- --'
 58T'-------- ------- ------------------------'
 101T'------------------------'/
*
READ WORK FILE 01 #USER-INPUT
 PERFORM FN1-PROCESS
END-WORK
*
DEFINE SUBROUTINE FN1-PROCESS
```

```
************************************************************************
* TO MAKE A FUNCTION 1 CALL:                                           *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                                *
*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                             *
*   (3) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD             *
*   (4) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER FIELD  *
*   (5) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD         *
*   (6) CALL GBI WITH 2 WORKAREAS                                      *
*   (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS                      *
************************************************************************
************************************************************************
* AS OF GEOSUPPORT 10.1,                                               *
*    TO RECEIVE ROADBED-SPECIFIC INFORMATION,                          *
*    SET THE ROADBED REQUEST SWITCH TO 'R', AS FOLLOWS:                *
*        MOVE 'R' TO GEO-WA1-IN-ROADBED-REQ-SWITCH.                    *
************************************************************************
 RESET GEOLW1
 MOVE '1 ' TO GEO-WA1-IN-FUNCTION-CODE
 MOVE #USER-BORO TO GEO-WA1-IN-BORO
 MOVE #USER-HSE-NUM TO GEO-WA1-IN-HOUSENUM
 MOVE #USER-STRT-NAME TO GEO-WA1-IN-STREET-1
*
 CALL 'GBI' W1NAT W2NAT
*
 IF GEO-WA1-OUT-RETURN-CODE NOT = '00' AND
    GEO-WA1-OUT-RETURN-CODE NOT = '01'
*
***** REPLACE YOUR OWN ERROR HANDLING ROUTINE HERE                 *****
*
  MOVE GEO-WA1-OUT-ERROR-MESSAGE TO #OUT-ERROR-MESSAGE-77
  WRITE NOTITLE
   1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
   49T '*** FUNCTION 1 GRC =' GEO-WA1-OUT-RETURN-CODE
   73T 'REASON CODE =' GEO-WA1-OUT-REASON-CODE /
   49T '***' #OUT-ERROR-MESSAGE-77 /
 ELSE
  IF GEO-WA1-OUT-RETURN-CODE = '01'
*
***** REPLACE YOUR OWN WARNING HANDLING ROUTINE HERE               *****
*
   MOVE GEO-WA1-OUT-ERROR-MESSAGE TO #OUT-ERROR-MESSAGE-77
   WRITE NOTITLE
    1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
    49T '*** FUNCTION 1 WARNING, GRC =' GEO-WA1-OUT-RETURN-CODE
    82T 'REASON CODE =' GEO-WA1-OUT-REASON-CODE /
    49T '***' #OUT-ERROR-MESSAGE-77
  END-IF
 END-IF
*
 IF GEO-WA1-OUT-RETURN-CODE = '00' OR
    GEO-WA1-OUT-RETURN-CODE = '01'
  MOVE GEO-WA1-OUT-RETURN-CODE TO #SAVE-RET-CODE
*
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR                    *****
***** PROCESSING SUCCESSFUL GEOSUPPORT FUNCTION 1 CALL             *****
*
************************************************************************
* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND              *
```

NATURAL SAMPLE PROGRAM #1 - Program Source Code - MSW (continued)

```
* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE                *
* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS                 *
* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D:             *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                                *
*   (2) SET WA1'S FUNCTION CODE FIELD TO D                             *
*   (3) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED                  *
*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE                 *
*              HAS SPACE FOR ONLY 25 CHARACTERS)                       *
*   (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN                  *
*       STREET NAMES FORMATTED FOR DISPLAY                             *
*   (5) MOVE WA2'S LOW PBSC FIELD TO WA1'S INPUT STREET CODE 1 FIELD   *
*   (6) MOVE WA2'S HIGH PBSC FIELD TO WA1'S INPUT STREET CODE 2 FIELD  *
*   (7) CALL GBI WITH 1 WORKAREA                                       *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS                      *
************************************************************************
  RESET GEOLW1
  MOVE 'D ' TO GEO-WA1-IN-FUNCTION-CODE
  MOVE 'C' TO GEO-WA1-IN-COMPACT-NAME-FLAG
  MOVE '25' TO GEO-WA1-IN-SNL
  MOVE GEO-WA2-FN1-LOW-PBSC(1) TO GEO-WA1-IN-STREETCODE-1
  MOVE GEO-WA2-FN1-HI-PBSC(1) TO GEO-WA1-IN-STREETCODE-2
*
  CALL 'GBI' W1NAT
*
  IF GEO-WA1-OUT-RETURN-CODE = '00'
   MOVE GEO-WA1-OUT-STREET-1 TO #OUT-STREET-1-SNL25
   MOVE GEO-WA1-OUT-STREET-2 TO #OUT-STREET-2-SNL25
*
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR                    *****
***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS                       *****
*
   IF #SAVE-RET-CODE = '01' RESET #USER-BORO
    #USER-HSE-NUM #USER-STRT-NAME
   END-IF
   WRITE NOTITLE
    1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
    49T GEO-WA2-FN1-ZIP 55T GEO-WA2-FN1-COMDIST-NUM
    58T GEO-WA2-FN1-POL-PRECINCT 67T GEO-WA2-FN1-SCHOOLDIST
    75T #OUT-STREET-1-SNL25 101T #OUT-STREET-2-SNL25 /
  ELSE
*
***** REPLACE YOUR OWN ERROR HANDLING ROUTINE HERE                 *****
*
   MOVE GEO-WA1-OUT-ERROR-MESSAGE TO #OUT-ERROR-MESSAGE-77
   WRITE NOTITLE
    1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
    49T '*** FUNCTION D GRC =' GEO-WA1-OUT-RETURN-CODE
    73T 'REASON CODE =' GEO-WA1-OUT-REASON-CODE /
    49T '***' #OUT-ERROR-MESSAGE-77 /
  END-IF
 END-IF
*
 END-SUBROUTINE
 END
```

NATURAL SAMPLE PROGRAM #1 - Program Source Code - COW

```
************************************************************************
* PGM NAME: GEOBUPGA    DATE: 08-18-98         MODIFIED : 08-28-06     *
*                                                                      *
* THIS PROGRAM MAKES FUNCTION 1 AND D CALLS TO GEOSUPPORT USING        *
* BORO, HOUSENUMBER, & STREET NAME SUPPLIED BY AN INSTREAM FILE.       *
* FUNCTION 1 RETURNS GEOGRAPHIC INFORMATION FOR AN ADDRESS.            *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.         *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  *
*          NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE       *
*                ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET     *
*                NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.         *
************************************************************************
*                                                                      *
************************************************************************
* USE OF GEOSUPPORT LDA (REFERENCED BELOW BY THE LOCAL USING STATEMENT)*
* IS STRONGLY ENCOURAGED.                                              *
************************************************************************
DEFINE DATA
 LOCAL USING GEOLP1
 LOCAL USING GEOLP2
*
***** REPLACE CODE BELOW WITH YOUR OWN INPUT FILE DECLARATION      *****
 LOCAL
  01 #USER-INPUT
     02 #USER-BORO                  (A1)
     02 #FILLER1                    (A1)
     02 #USER-HSE-NUM               (A12)
     02 #FILLER2                    (A1)
     02 #USER-STRT-NAME             (A32)
     02 #FILLER3                    (A33)
*
  01 #SAVE-RET-CODE                 (A2)
  01 #OUT-STREET-1-SNL25            (A25)
  01 #OUT-STREET-2-SNL25            (A25)
  01 #OUT-ERROR-MESSAGE-77          (A77)
*
  01 #B5SC (A6)
  01 REDEFINE #B5SC
     02 #B5SC-BORO (A1)
     02 #B5SC-5SC (A5)
*
END-DEFINE
*
FORMAT LS=133 PS=65
*
***** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT               *****
WRITE NOTITLE
 1T'SAMPLE NATURAL PROGRAM #1 EXECUTION OUTPUT'//
 1T'*****---------- INPUT ADDRESS ------------*****'
 49T'*****--------------------- SELECTED OUTPUT'
 92T'ITEMS ----------------------*****'//
 1T'B HOUSE NUMBER IN-STREET-NAME                     ZIP CD'
 58T'NYPD-PCT SCHLDST LOW CROSS STREET         '
 101T'HIGH CROSS STREET        '/
 1T'- ------------ -------------------------------- ----- --'
 58T'-------- ------- ------------------------'
 101T'------------------------'/
*
```

NATURAL SAMPLE PROGRAM #1 - Program Source Code - COW (continued)

```
READ WORK FILE 01 #USER-INPUT
 PERFORM FN1-PROCESS
END-WORK
*
DEFINE SUBROUTINE FN1-PROCESS
************************************************************************
* TO MAKE A FUNCTION 1 CALL:                                           *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                                *
*   (2) SET WA1'S FUNCTION CODE FIELD TO 1                             *
*   (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)          *
*       TO USE CHARACTER-ONLY WORK AREA (COWS)                         *
*   (4) MOVE THE INPUT BORO TO WA1'S INPUT BORO CODE FIELD             *
*   (5) MOVE THE INPUT HOUSE NUMBER TO WA1'S INPUT HOUSE NUMBER FIELD  *
*   (6) MOVE THE INPUT STREET TO WA1'S INPUT STREET NAME FIELD         *
*   (7) CALL GBI WITH 2 WORKAREAS                                      *
*   (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS                      *
************************************************************************
************************************************************************
* AS OF GEOSUPPORT 10.1,                                               *
*    TO RECEIVE ROADBED-SPECIFIC INFORMATION,                          *
*    SET THE ROADBED REQUEST SWITCH TO 'R', AS FOLLOWS:                *
*        MOVE 'R' TO PIWA1-IN-ROADBED-REQ-SWITCH.                      *
************************************************************************
 RESET GEOLP1
 MOVE '1 ' TO PIWA1-IN-FUNCTION-CODE
 MOVE 'C' TO PIWA1-IN-PLATFORM-INDICATOR
 MOVE #USER-BORO TO PIWA1-IN-BORO-1
 MOVE #USER-HSE-NUM TO PIWA1-IN-HOUSENUM-DISPLAY
 MOVE #USER-STRT-NAME TO PIWA1-IN-STREET-1
*
 CALL 'GBI' P1NAT P2NAT
*
 IF PIWA1-OUT-RETURN-CODE NOT = '00' AND
    PIWA1-OUT-RETURN-CODE NOT = '01'
*
***** REPLACE YOUR OWN ERROR HANDLING ROUTINE HERE                 *****
*
  MOVE PIWA1-OUT-ERROR-MESSAGE TO #OUT-ERROR-MESSAGE-77
  WRITE NOTITLE
   1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
   49T '*** FUNCTION 1 GRC =' PIWA1-OUT-RETURN-CODE
   73T 'REASON CODE =' PIWA1-OUT-REASON-CODE /
   49T '***' #OUT-ERROR-MESSAGE-77 /
 ELSE
  IF PIWA1-OUT-RETURN-CODE = '01'
*
***** REPLACE YOUR OWN WARNING HANDLING ROUTINE HERE               *****
*
   MOVE PIWA1-OUT-ERROR-MESSAGE TO #OUT-ERROR-MESSAGE-77
   WRITE NOTITLE
    1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
    49T '*** FUNCTION 1 WARNING, GRC =' PIWA1-OUT-RETURN-CODE
    82T 'REASON CODE =' PIWA1-OUT-REASON-CODE /
    49T '***' #OUT-ERROR-MESSAGE-77
  END-IF
 END-IF
*
 IF PIWA1-OUT-RETURN-CODE = '00' OR
```

```
    PIWA1-OUT-RETURN-CODE = '01'
  MOVE PIWA1-OUT-RETURN-CODE TO #SAVE-RET-CODE
*
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR                       *****
***** PROCESSING SUCCESSFUL GEOSUPPORT FUNCTION 1 CALL                *****
*
**************************************************************************
* THIS PROGRAM ASSUMES THERE EXISTS AT LEAST ONE HIGH AND                *
* ONE LOW CROSS STREET.  TO GET THE STREET NAMES OF THE                  *
* FIRST-LISTED HIGH AND FIRST-LISTED LOW CROSS STREETS                   *
* FROM THE HIGH AND LOW STREET CODE LISTS CALL FUNCTION D:               *
*   (1) INITIALIZE WORKAREA 1 TO SPACES                                  *
*   (2) SET WA1'S FUNCTION CODE FIELD TO D                               *
*   (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)            *
*       TO USE CHARACTER-ONLY WORK AREA (COWS)                           *
*   (4) SET WA1'S STREET NAME LENGTH FIELD TO DESIRED                    *
*       VALUE (IN THIS CASE 25 BECAUSE THE REPORT LINE                   *
*              HAS SPACE FOR ONLY 25 CHARACTERS)                         *
*   (5) USE THE COMPACT STREET NAMES OPTION TO OBTAIN                    *
*       STREET NAMES FORMATTED FOR DISPLAY                               *
*   (6) MOVE WA2'S LOW B5SC FIELD TO WA1'S INPUT STREET CODE 1 FIELD     *
*   (7) MOVE WA2'S HIGH B5SC FIELD TO WA1'S INPUT STREET CODE 2 FIELD    *
*   (8) CALL GBI WITH 1 WORKAREA                                         *
*   (9) CHECK RETURN CODES FOR ERRORS OR WARNINGS                        *
**************************************************************************
  RESET GEOLP1
  MOVE 'C' TO PIWA1-IN-PLATFORM-INDICATOR
  MOVE 'D ' TO PIWA1-IN-FUNCTION-CODE
  MOVE '25' TO PIWA1-IN-SNL
  MOVE 'C' TO PIWA1-IN-SN-NORM-FORMAT
  MOVE PIWA2-FN1-LOW-B5SC(1) TO #B5SC
  MOVE #B5SC-BORO TO PIWA1-IN-BORO-1
  MOVE #B5SC-5SC TO PIWA1-IN-10SC-1
  MOVE PIWA2-FN1-HI-B5SC(1) TO #B5SC
  MOVE #B5SC-BORO TO PIWA1-IN-BORO-2
  MOVE #B5SC-5SC TO PIWA1-IN-10SC-2
*
  CALL 'GBI' P1NAT
*
  IF PIWA1-OUT-RETURN-CODE = '00'
   MOVE PIWA1-OUT-STREET-1 TO #OUT-STREET-1-SNL25
   MOVE PIWA1-OUT-STREET-2 TO #OUT-STREET-2-SNL25
*
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR                       *****
***** PROCESSING SUCCESSFUL GEOSUPPORT CALLS                          *****
*
   IF #SAVE-RET-CODE = '01' RESET #USER-BORO
    #USER-HSE-NUM #USER-STRT-NAME
   END-IF
   WRITE NOTITLE
    1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
    49T PIWA2-FN1-ZIP 55T PIWA2-FN1-COM-DIST-NUM
    58T PIWA2-FN1-POL-PRECINCT 67T PIWA2-FN1-SCHL-DIST
    75T #OUT-STREET-1-SNL25 101T #OUT-STREET-2-SNL25 /
  ELSE
*
***** REPLACE YOUR OWN ERROR HANDLING ROUTINE HERE                    *****
*
```

NATURAL SAMPLE PROGRAM #1 - Program Source Code - COW (continued)

```
   MOVE PIWA1-OUT-ERROR-MESSAGE TO #OUT-ERROR-MESSAGE-77
   WRITE NOTITLE
    1T #USER-BORO 3T #USER-HSE-NUM 16T #USER-STRT-NAME
    49T '*** FUNCTION D GRC =' PIWA1-OUT-RETURN-CODE
    73T 'REASON CODE =' PIWA1-OUT-REASON-CODE /
    49T '***' #OUT-ERROR-MESSAGE-77 /
  END-IF
 END-IF
*
 END-SUBROUTINE
 END
```

NATURAL SAMPLE PROGRAM #1 - Job Stream

```
//GEOBUPG1 JOB    YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  NATURAL SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #1   ***
//***                     MSW FORMAT                                 ***
//*********************************************************************
//S1  EXEC NT3MPM1M,REGION=7000K
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                                 *//
//*  AS OF GEOSUPPORT VERSION 10.0,                                 *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP       *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATSETS:           *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                             *//
//*       A030.GEO.SUPPORT.LOADLIB                                  *//
//*                                                                 *//
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//NAT.STEPLIB DD
//            DD
//            DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//SYSPRINT    DD SYSOUT=A
//SYSOUT      DD SYSOUT=A,DCB=(LRECL=132)
//CMPRINT     DD SYSOUT=A,DCB=(LRECL=132)
//CMWKF01     DD *
1,22          ,READE ST
1,500         ,DUANE ST
1,2-4         ,BROADWAY
4,165-100     ,BAISLEY BLVD
4,165-1000    ,BAISLEY BLVD
//CMSYNIN     DD *
Your_Application_ID,Your-User_ID
%*
Your-Password
L L GEOLW1     [For COW: GEOLP1]
L L GEOLW2     [For COW: GEOLP2]
L P GEOBUPG1   [For COW Sample:  GEOBUPGA]
GEOBUPG1       [For COW Sample:  GEOBUPGA}
FIN
//INCLIB      DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                                 *//
//*  AS OF GEOSUPPORT VERSION 10.0,                                 *//
//*  DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,       *//
//*  ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT          *//
//*  IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.         *//
//*  TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.        *//
//*                                                                 *//
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//SYSUDUMP    DD DUMMY
//
```

NATURAL SAMPLE PROGRAM #1 - Output Report

```
SAMPLE NATURAL PROGRAM #1 EXECUTION OUTPUT

*****---------- INPUT ADDRESS -----------***** *****-------------------- SELECTED OUTPUT ITEMS ----------------------*****

B HOUSE NUMBER IN-STREET-NAME                    ZIP CD NYPD-PCT SCHLDST LOW CROSS STREET         HIGH CROSS STREET
- ------------ -------------------------------- ----- -- -------- ------- ------------------------ ------------------------

1 22           READE ST                         10007 01 005      02      ELK STREET               BROADWAY

1 500          DUANE ST                         *** FUNCTION 1 GRC = 42 REASON CODE =
                                                *** ADDRESS NUMBER OUT OF RANGE

1 2-4          BROADWAY                         *** FUNCTION 1 WARNING, GRC = 01 REASON CODE = 1
                                                *** ADDR NUMBER ALTERED: RANGE ASSUMED. USING DIGITS BEFORE DASH ONLY
                                                10004 01 001      02      STONE STREET             BOWLING GREEN

4 165-100      BAISLEY BLVD                     11434 12 113      28      GUY R BREWER BOULEVARD   BEDELL STREET

4 165-1000     BAISLEY BLVD                     *** FUNCTION 1 GRC = 13 REASON CODE = 2
                                                *** ADDRESS NBR 165-1000  HAS MORE THAN 3 DIGITS AFTER THE DASH.
```

NATURAL SAMPLE PROGRAM #2

- Program Source Code - MSW

- Program Source Code - COW

- Input Job Stream

- Output Report

NATURAL SAMPLE PROGRAM #2 - Program Source Code - MSW

```
************************************************************************
* PGM NAME: GEOBUPG2    DATE: 08-18-98                                 *
*                                                                      *
* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING TWO    *
* BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.             *
* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION.       *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.         *
************************************************************************
*         NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE        *
*               ORIGINAL CALL TO FUNCTION 1, ALL THE CROSS STREET      *
*               NAMES WOULD HAVE BEEN RETURNED BY FUNCTION 1.          *
************************************************************************
*
************************************************************************
* USE OF GEOSUPPORT LDA (REFERENCED BELOW BY THE LOCAL USING STATEMENT)*
* IS STRONGLY ENCOURAGED.                                              *
************************************************************************
*
DEFINE DATA
 LOCAL USING GEOLW1
 LOCAL USING GEOLW2
*
***** REPLACE CODE BELOW WITH YOUR OWN INPUT DATA DECLARATION      *****
*
 LOCAL
  01 #USER-INPUT
     02 #USER-BORO1                  (A1)
     02 #FILLER1                     (A1)
     02 #USER-STRT-NAME1             (A32)
     02 #FILLER2                     (A1)
     02 #USER-BORO2                  (A1)
     02 #FILLER3                     (A1)
     02 #USER-STRT-NAME2             (A32)
     02 #FILLER4                     (A11)
*
  01 #INDEX                          (I1)
*
END-DEFINE
*
FORMAT LS=133 PS=65
*
***** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT               *****
*
WRITE NOTITLE
 1T'SAMPLE NATURAL PROGRAM #2 EXECUTION OUTPUT'//
 1T'*****----------------- INPUT INTERSECTION'
 43T'----------------------*****'
 71T'*****------------ SELECTED OUTPUT ITEMS -------------*****'//
 1T 'B IN-STREET-NAME-1                '
 36T'B IN-STREET-NAME-2                '
 71T'  ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES'/
 1T '- --------------------------------'
 36T'- --------------------------------'
 71T'----- -- -------- ------- --------------------------------'
*
READ WORK FILE 01 #USER-INPUT
 PERFORM FN2-PROCESS
END-WORK
```

NATURAL SAMPLE PROGRAM #2 - Program Source Code - MSW (continued)

```
*
DEFINE SUBROUTINE FN2-PROCESS
***********************************************************************
* TO MAKE A FUNCTION 2 CALL:                                          *
*    (1) INITIALIZE WORKAREA 1 TO SPACES                              *
*    (2) SET WA1'S FUNCTION-CODE TO 2                                 *
*    (3) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD       *
*    (4) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME FIELD   *
*    (5) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD     *
*    (6) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2 FIELD *
*    (7) CALL GBI WITH 2 WORKAREAS                                    *
*    (8) CHECK RETURN CODES FOR ERRORS OR WARNINGS                    *
***********************************************************************
 RESET GEOLW1
 MOVE '2 ' TO GEO-WA1-IN-FUNCTION-CODE
 MOVE #USER-BORO1 TO GEO-WA1-IN-BORO
 MOVE #USER-STRT-NAME1 TO GEO-WA1-IN-STREET-1
 MOVE #USER-BORO2 TO GEO-WA1-IN-BORO-2
 MOVE #USER-STRT-NAME2 TO GEO-WA1-IN-STREET-2
*
 CALL 'GBI' W1NAT W2NAT
*
 IF GEO-WA1-OUT-RETURN-CODE NOT = '00' AND
    GEO-WA1-OUT-RETURN-CODE NOT = '01'
*
***** REPLACE CODE BELOW WITH YOUR OWN ERROR HANDLING ROUTINE HERE *****
*
  WRITE NOTITLE /
   1T '***** FUNCTION 2 GRC =' GEO-WA1-OUT-RETURN-CODE
   27T 'REASON CODE ='GEO-WA1-OUT-REASON-CODE
   43T ','GEO-WA1-OUT-ERROR-MESSAGE /
   1T #USER-BORO1 3T #USER-STRT-NAME1
   36T #USER-BORO2 38T #USER-STRT-NAME2
 ELSE
  IF GEO-WA1-OUT-RETURN-CODE = '01'
*
*** REPLACE CODE BELOW WITH YOUR OWN WARNING HANDLING ROUTINE HERE   ***
*
   WRITE NOTITLE /
    1T '***** FUNCTION 2 WARNING, GRC =' GEO-WA1-OUT-RETURN-CODE
    37T 'REASON CODE =' GEO-WA1-OUT-REASON-CODE
    53T ','GEO-WA1-OUT-ERROR-MESSAGE /
    1T #USER-BORO1 3T #USER-STRT-NAME1
    36T #USER-BORO2 38T #USER-STRT-NAME2
  END-IF
 END-IF
*
  IF GEO-WA1-OUT-RETURN-CODE = '00' OR
     GEO-WA1-OUT-RETURN-CODE = '01'
*
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR                   *****
***** PROCESSING SUCCESSFUL GEOSUPPORT FUNCTION 2 CALL            *****
*
   FOR #INDEX 1 TO GEO-WA2-FN2-NUM-OF-INTERSECTS
**********************************************************************
* TO GET STREET NAMES FOR INTERSECTING STREET CODES                  *
* MAKE A FUNCTION D CALL:                                            *
*    (1) INITIALIZE WORKAREA 1 TO SPACES                             *
```

NATURAL SAMPLE PROGRAM #2 - Program Source Code - MSW (continued)

```
*     (2) SET THE WA1'S FUNCTION CODE FIELD TO D                        *
*     (3) USE THE COMPACT STREET NAMES OPTION TO OBTAIN                 *
*         STREET NAMES FORMATTED FOR DISPLAY                            *
*     (4) MOVE THE PACKED BORO AND STREET CODE TO                       *
*         WA1'S INPUT STREET CODE 1 FIELD                               *
*     (5) CALL GBI WITH 1 WORKAREA                                      *
*     (6) CHECK RETURN CODES FOR ERRORS OR WARNINGS                     *
***********************************************************************
    RESET GEOLW1
    MOVE 'D ' TO GEO-WA1-IN-FUNCTION-CODE
    MOVE 'C ' TO GEO-WA1-IN-COMPACT-NAME-FLAG
    MOVE '25' TO GEO-WA1-IN-SNL
    MOVE GEO-WA2-FN2-INTERSECT-PBSC(#INDEX) TO GEO-WA1-IN-STREETCODE-1
*
    CALL 'GBI' W1NAT
*
    IF GEO-WA1-OUT-RETURN-CODE = '00'
*
***** INSERT YOUR OWN CODE HERE FOR                                *****
***** PROCESSING SUCCESSFUL FUNCTION D CALLS                       *****
*
     IF #INDEX = 1
      WRITE NOTITLE /
       1T #USER-BORO1 3T #USER-STRT-NAME1
       36T #USER-BORO2 38T #USER-STRT-NAME2
       71T GEO-WA2-FN2-ZIP 77T GEO-WA2-FN2-COMDIST-NUM
       80T GEO-WA2-FN2-POL-PRECINCT 89T GEO-WA2-FN2-SCHOOLDIST
       97T GEO-WA1-OUT-STREET-1
     ELSE
      WRITE NOTITLE
       97T GEO-WA1-OUT-STREET-1
     END-IF
    ELSE IF GEO-WA1-OUT-RETURN-CODE NOT = '00'
*
***** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE                  *****
*
     WRITE NOTITLE /
      1T '***** FUNCTION D GRC =' GEO-WA1-OUT-RETURN-CODE
      27T 'REASON CODE =' GEO-WA1-OUT-REASON-CODE
      43T ','GEO-WA1-OUT-ERROR-MESSAGE /
      1T #USER-BORO1 3T #USER-STRT-NAME1
      36T #USER-BORO2 38T #USER-STRT-NAME2
    END-IF
    END-IF
   END-FOR
 END-IF
*
END-SUBROUTINE
END
```

NATURAL SAMPLE PROGRAM #2 - Program Source Code - COW

```
************************************************************************
* PGM NAME: GEOBUPGB    DATE: 08-18-98     MODIFIED : 08-28-06         *
*                                                                      *
* THIS PROGRAM MAKES FUNCTION 2 AND D CALLS TO GEOSUPPORT USING TWO    *
* BOROS AND TWO STREET NAMES SUPPLIED BY AN INSTREAM FILE.             *
* FUNCTION 2 RETURNS GEOGRAPHIC INFORMATION FOR AN INTERSECTION.       *
* FUNCTION D TRANSLATES AN INPUT STREET CODE TO A STREET NAME.         *
************************************************************************
*         NOTE: IF THE CROSS STREET NAMES FLAG WERE USED IN THE       *
*               ORIGINAL CALL TO FUNCTION 2, ALL THE CROSS STREET     *
*               NAMES WOULD HAVE BEEN REUTRNED BY FUNCTION 2.         *
************************************************************************
*                                                                      *
************************************************************************
* USE OF GEOSUPPORT LDA (REFERENCED BELOW BY THE LOCAL USING STATEMENT)*
* IS STRONGLY ENCOURAGED.                                              *
************************************************************************
*
DEFINE DATA
 LOCAL USING GEOLP1
 LOCAL USING GEOLP2
*
***** REPLACE CODE BELOW WITH YOUR OWN INPUT DATA DECLARATION      *****
*
 LOCAL
  01 #USER-INPUT
     02 #USER-BORO1                  (A1)
     02 #FILLER1                     (A1)
     02 #USER-STRT-NAME1             (A32)
     02 #FILLER2                     (A1)
     02 #USER-BORO2                  (A1)
     02 #FILLER3                     (A1)
     02 #USER-STRT-NAME2             (A32)
     02 #FILLER4                     (A11)
*
  01 #INDEX                          (I1)
*
  01 #B5SC (A6)
  01 REDEFINE #B5SC
     02 #B5SC-BORO (A1)
     02 #B5SC-5SC (A5)
*
  01 #NUM-INTERSECT-A (A1)
  01 REDEFINE #NUM-INTERSECT-A
     02 #NUM-INTERSECT-N (N1)
*
END-DEFINE
*
FORMAT LS=133 PS=60
*
***** REPLACE CODE BELOW WITH YOUR OWN REPORT LAYOUT               *****
*
WRITE NOTITLE
 1T'SAMPLE NATURAL PROGRAM #2 EXECUTION OUTPUT'//
 1T'*****----------------- INPUT INTERSECTION'
 43T'----------------------*****'
 71T'*****------------- SELECTED OUTPUT ITEMS -------------*****'//
 1T 'B IN-STREET-NAME-1                '
```

NATURAL SAMPLE PROGRAM #2 - Program Source Code - COW (continued)

```
 36T'B IN-STREET-NAME-2                '
 71T'  ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES'/
 1T '- ------------------------------'
 36T'- ------------------------------'
 71T'----- -- -------- ------- ------------------------------'
*
READ WORK FILE 01 #USER-INPUT
 PERFORM FN2-PROCESS
END-WORK
*
DEFINE SUBROUTINE FN2-PROCESS
***********************************************************************
* TO MAKE A FUNCTION 2 CALL:                                          *
*    (1) INITIALIZE WORKAREA 1 TO SPACES                              *
*    (2) SET WA1'S FUNCTION-CODE TO 2                                 *
*    (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAINFRAME)        *
*        TO USE CHARACTER ONLY WORK AREAS (COWS)                      *
*    (4) MOVE THE 1ST INPUT BORO TO WA1'S INPUT BORO CODE FIELD       *
*    (5) MOVE THE 1ST INPUT STREET TO WA1'S INPUT STREET NAME FIELD   *
*    (6) MOVE THE 2ND INPUT BORO TO WA1'S INPUT BORO CODE 2 FIELD     *
*    (7) MOVE THE 2ND INPUT STREET TO WA1'S INPUT STREET NAME 2 FIELD  *
*    (8) CALL GBI WITH 2 WORKAREAS                                    *
*    (9) CHECK RETURN CODES FOR ERRORS OR WARNINGS                    *
***********************************************************************
 RESET GEOLP1
 MOVE '2 ' TO PIWA1-IN-FUNCTION-CODE
 MOVE 'C' TO PIWA1-IN-PLATFORM-INDICATOR
 MOVE #USER-BORO1 TO PIWA1-IN-BORO-1
 MOVE #USER-STRT-NAME1 TO PIWA1-IN-STREET-1
 MOVE #USER-BORO2 TO PIWA1-IN-BORO-2
 MOVE #USER-STRT-NAME2 TO PIWA1-IN-STREET-2
*
 CALL 'GBI' P1NAT P2NAT
*
 IF PIWA1-OUT-RETURN-CODE NOT = '00' AND
    PIWA1-OUT-RETURN-CODE NOT = '01'
*
***** REPLACE CODE BELOW WITH YOUR OWN ERROR HANDLING ROUTINE HERE *****
*
  WRITE NOTITLE /
   1T '***** FUNCTION 2 GRC =' PIWA1-OUT-RETURN-CODE
   27T 'REASON CODE ='PIWA1-OUT-REASON-CODE
   43T ','PIWA1-OUT-ERROR-MESSAGE /
   1T #USER-BORO1 3T #USER-STRT-NAME1
   36T #USER-BORO2 38T #USER-STRT-NAME2
 ELSE
  IF PIWA1-OUT-RETURN-CODE = '01'
*
*** REPLACE CODE BELOW WITH YOUR OWN WARNING HANDLING ROUTINE HERE   ***
*
   WRITE NOTITLE /
    1T '***** FUNCTION 2 WARNING, GRC =' PIWA1-OUT-RETURN-CODE
    37T 'REASON CODE =' PIWA1-OUT-REASON-CODE
    53T ','PIWA1-OUT-ERROR-MESSAGE /
    1T #USER-BORO1 3T #USER-STRT-NAME1
    36T #USER-BORO2 38T #USER-STRT-NAME2
  END-IF
 END-IF
```

```
*
  IF PIWA1-OUT-RETURN-CODE = '00' OR
     PIWA1-OUT-RETURN-CODE = '01'
*
***** REPLACE CODE BELOW WITH YOUR OWN CODE FOR                    *****
***** PROCESSING SUCCESSFUL GEOSUPPORT FUNCTION 2 CALL             *****
*
  MOVE PIWA2-FN2-NUM-OF-INTERSECTS TO #NUM-INTERSECT-A
  FOR #INDEX 1 TO #NUM-INTERSECT-N
************************************************************************
* TO GET STREET NAMES FOR INTERSECTING STREET CODES                    *
* MAKE A FUNCTION D CALL:                                              *
*    (1) INITIALIZE WORKAREA 1 TO SPACES                               *
*    (2) SET THE WA1'S FUNCTION CODE FIELD TO D                        *
*    (3) SET THE PLATFORM INDICATOR SWITCH (NON-IBM-MAIN-FRAME)        *
*        TO USE CHARACTER ONLY WORK AREAS (COWS)                       *
*    (4) USE THE COMPACT STREET NAMES OPTION TO OBTAIN                 *
*        STREET NAMES FORMATTED FOR DISPLAY                            *
*    (5) MOVE THE BORO AND STREET CODE TO                              *
*        WA1'S INPUT STREET CODE 1 FIELD                               *
*    (6) CALL GBI WITH 1 WORKAREA                                      *
*    (7) CHECK RETURN CODES FOR ERRORS OR WARNINGS                     *
************************************************************************
    RESET GEOLP1
    MOVE 'D ' TO PIWA1-IN-FUNCTION-CODE
    MOVE 'C' TO PIWA1-IN-PLATFORM-INDICATOR
    MOVE 'C' TO PIWA1-IN-SN-NORM-FORMAT
*   MOVE PIWA2-FN2-INTERSECT-B5SC(#INDEX) TO PIWA1-IN-10SC-1
    MOVE PIWA2-FN2-INTERSECT-B5SC(#INDEX) TO #B5SC
    MOVE #B5SC-BORO TO PIWA1-IN-BORO-1
    MOVE #B5SC-5SC TO PIWA1-IN-10SC-1
*
    CALL 'GBI' P1NAT
*
    IF PIWA1-OUT-RETURN-CODE = '00'
*
***** INSERT YOUR OWN CODE HERE FOR                                *****
***** PROCESSING SUCCESSFUL FUNCTION D CALLS                       *****
*
     IF #INDEX = 1
      WRITE NOTITLE /
       1T #USER-BORO1 3T #USER-STRT-NAME1
       36T #USER-BORO2 38T #USER-STRT-NAME2
       71T PIWA2-FN2-ZIP 77T PIWA2-FN2-COM-DIST-NUM
       80T PIWA2-FN2-POL-PRECINCT 89T PIWA2-FN2-SCHL-DIST
       97T PIWA1-OUT-STREET-1
     ELSE
      WRITE NOTITLE
       97T PIWA1-OUT-STREET-1
     END-IF
    ELSE IF PIWA1-OUT-RETURN-CODE NOT = '00'
*
***** INSERT YOUR OWN ERROR HANDLING ROUTINE HERE                  *****
*
     WRITE NOTITLE /
      1T '***** FUNCTION D GRC =' PIWA1-OUT-RETURN-CODE
      27T 'REASON CODE =' PIWA1-OUT-REASON-CODE
      43T ','PIWA1-OUT-ERROR-MESSAGE /
```

NATURAL SAMPLE PROGRAM #2 - Program Source Code - COW (continued)

```
      1T #USER-BORO1 3T #USER-STRT-NAME1
      36T #USER-BORO2 38T #USER-STRT-NAME2
    END-IF
    END-IF
   END-FOR
 END-IF
*
END-SUBROUTINE
END
```

NATURAL SAMPLE PROGRAM #2 - Job Stream

```
//GEOBUPG2 JOB    YOUR-JOB-CARD-INFORMATION
//*
//*********************************************************************
//***  NATURAL SAMPLE BATCH GEOSUPPORT USER APPLICATION PROGRAM #2   ***
//***                        MSW FORMAT                              ***
//*********************************************************************
//S1  EXEC NT3MPM1M,REGION=7000K
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                                 *//
//*  AS OF GEOSUPPORT VERSION 10.0,                                 *//
//*  THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP       *//
//*  MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATSETS:           *//
//*       A030.GEO.SUPPORT.PDSE.LOADLIB                             *//
//*       A030.GEO.SUPPORT.LOADLIB                                  *//
//*                                                                 *//
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//NAT.STEPLIB DD
//            DD
//            DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR
//            DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//SYSPRINT    DD SYSOUT=A
//SYSOUT      DD SYSOUT=A,DCB=(LRECL=132)
//CMPRINT     DD SYSOUT=A,DCB=(LRECL=132)
//CMWKF01     DD *
1,CHAMBERS ST                     ,1,HUDSON ST
1,SIXTH AV                        ,1,W. 8 ST
1,DUANE ST                        ,1,READE ST
//CMSYNIN     DD *
Your-Application-ID,Your-User-ID
%*
Your-Password
L L GEOLW1         [For COW: GEOLP1]
L L GEOLW2         [For COW: GEOLP2]
L P GEOBUPG2       [For COW Sample:  GEOBUPGB]
GEOBUPG2           [For COW Sample:  GEOBUPGB]
FIN
//INCLIB      DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//*                                                                 *//
//*  AS OF GEOSUPPORT VERSION 10.0,                                 *//
//*  DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD,       *//
//*  ETC) ARE NO LONGER NEEDED AND ARE IGNORED. GEOSUPPORT          *//
//*  IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.         *//
//*  TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.        *//
//*                                                                 *//
//* * * * * * *  * * * * * * * * * * * * * * * * * * * * * * * * * *//
//SYSUDUMP    DD DUMMY
//
```

NATURAL SAMPLE PROGRAM #2 - Output Report

```
SAMPLE NATURAL PROGRAM #2 EXECUTION OUTPUT

*****---------------- INPUT INTERSECTION ----------------------***** *****------------ SELECTED OUTPUT ITEMS -------------*****

B IN-STREET-NAME-1                 B IN-STREET-NAME-2                   ZIP CD NYPD-PCT SCHLDST INTERSECTING STREET NAMES
- -------------------------------- - -------------------------------- ----- -- -------- ------- --------------------------------

1 CHAMBERS ST                      1 HUDSON ST                        10007 01 001      02      CHAMBERS STREET
                                                                                                HUDSON STREET
                                                                                                WEST BROADWAY

1 SIXTH AV                         1 W. 8 ST                          10014 02 006      02      6 AVENUE
                                                                                                GREENWICH AVENUE
                                                                                                WEST 8 STREET

***** FUNCTION 2 GRC = 62 REASON CODE =   , READE STREET & DUANE STREET DO NOT INTERSECT
1 DUANE ST                         1 READE ST
```

APPENDIX 9: GBAT REFERENCE TABLES

Table A9-1: GBAT Control Entry Descriptions by Keyword

This table lists all of the control entries alphabetically by keyword. Each control entry's coding format is indicated, and its purpose and usage are described. Control entry variables are indicated using 'S' and 'L' to represent the starting position and length, respectively, of a field in the input data records, and 'V' to represent other types of variables. Certain control entries do not have full table entries of their own but are cross-referenced to closely related control entries.

Control Entry	Description
ALIASES=V	Specifies whether an input alias file is to be used during this GBAT run. If so, during the processing of the input data file, the user-defined street name aliases in the alias file supplement the set of street names that Geosupport recognizes (see Section IX.6). This control entry is optional; if it is not coded, the default value is NO. The valid variable values are NO, YES and VAL. ALIASES=NO directs GBAT not to perform any alias processing. If there is an ALIASES DD statement in the JCL, it is ignored. ALIASES=YES directs GBAT to validate the ALIASES file, and then to process the input data file whether or not there are any invalid records in the alias file. An ALIASES DD statement referring to the alias file is required to be in the JCL. ALIASES=VAL directs GBAT to validate the alias file, and then to process the input data file only if there are no invalid records in the alias file. An ALIASES DD statement referring to the alias file is required to be in the JCL.
BBL=S,10	Specifies the starting position and length of the input BBL field in the input data records. This control entry is valid only for Function BL. This control entry can be used in place of the three control entries BORO, BLOCK and LOT whenever the input data records contain those three items in adjacent positions so that they can be treated collectively as a BBL field. The length value must be explicitly coded as '10'.
BIN=S or **BIN=S,7**	Specifies the starting position and length of the input Building Identification Number (BIN) field in the input data records. This control entry is valid only for Function BN. An input BIN field must always have a length of seven bytes, which may be coded explicitly as '7' in this control entry or it may be left uncoded, in which case it defaults to that value.
BLOCK=S or **BLOCK=S,5**	Specifies the starting position and length of the input tax block field in the input data records. This control entry is valid only for Function BL. Either all three control entries BORO, BLOCK and LOT must be specified, or the control entry BBL must be specified. An input tax block field must always have a length of five bytes, which may be coded explicitly as '5' in this control entry, or it may be left uncoded, in which case it defaults to that value.

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description
BORO=S,L	Specifies the starting position and length of the input borough code field in the input data records. This control entry is valid for all functions except Function BN. It is mandatory for functions that require an input borough code field. For Functions 2, 3, 3C, 3S and D, which accept multiple input street fields, the field specified by BORO serves as the input borough code field for the input street field specified by the control entry ONSTREET or STRTCODE; in addition, if the control entries CROSSBORO1 and CROSSBORO2 are not coded, it also serves as the input borough code field for the other input street fields. The maximum permissible length value of BORO is L=12. Note: The input borough code field specified by BORO may contain user-defined, non-standard borough code values - see discussion at table entry for BRONX.
BRONX=V **BROOKLYN=V** **MANHATTAN=V** **QUEENS=V** **STATEN=V**	GBAT can accept non-standard, user-defined borough code values in the input borough code fields specified by the control entries BORO, CROSSBORO1 and CROSSBORO2. (If there is more than one input borough code field, the same borough code values must be used in all of them.) The five control entries BRONX, BROOKLYN, MANHATTAN, QUEENS and STATEN are used to specify the character strings that represent each borough in those input borough code fields. (Note: these control entries do not pertain to the borough code sub-fields that are imbedded within larger data items such as BBL, BIN and B7SC, which must always contain the standard Geosupport borough code values.) The five borough code values may be specified as any strings of non-blank characters the lengths of which do not exceed the length value specified in the BORO control entry (which has a maximum permissible length value of 12). Imbedded blanks should not be included in user-defined borough code values, since the first blank that GBAT encounters when scanning a borough code value terminates the scan. For example, coding STATEN=STATEN ISLAND would cause GBAT to interpret the control entry as STATEN=STATEN and to interpret ISLAND as the next keyword in the control file. Since ISLAND is an invalid control keyword, GBAT would terminate abnormally. However, coding STATEN=STATENISLAND (without the imbedded blank) is acceptable. These five control entries are optional, but if any of them is coded, all five must be coded. If these control entries are not coded, the default values are the standard Geosupport borough codes, as follows: MANHATTAN=1 BRONX=2 BROOKLYN=3 QUEENS=4 STATEN=5
BROOKLYN	See BRONX.
B7SC1=S or	Specify the starting positions of up to three input Borough-and-7-digit Street Code

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

<u>Control Entry</u>	<u>Description</u>
B7SC1=S,8 **B7SC2=S** or **B7SC2=S,8** **B7SC3=S** or **B7SC3=S,8**	(B7SC) fields for input to Function DG. An input B7SC field must always have a length of 8, which may be coded explicitly in these control entries, or it may be left uncoded, in which case it defaults to 8.
B10SC1=S or **B10SC1=S,11** **B10SC2=S** or **B10SC2=S,11** **B10SC3=S** or **B10SC3=S,11**	Specify the starting positions of up to three input Borough-and-10-digit Street Code (B10SC) fields for input to Function DN. An input B10SC field must always have a length of 11, which may be coded explicitly in these control entries, or it may be left uncoded, in which case it defaults to 11.
COMPACT=V	Specifies whether the Compact Names option (described in Section III.3) is in effect. This control entry is valid only for functions that return normalized street name output. For such functions, this control entry is optional, and NO is the default value. COMPACT=YES specifies that the Compact Names option is in effect. Street names with numeric components are normalized into the compact format. COMPACT=NO specifies that the Compact Names option is not in effect. Street names with numeric components are normalized into the sort format.
COMPASS=S	Specifies the position of an input compass direction field in the input data records. This control entry is never coded with a length variable; GBAT always assumes an input compass direction field to be one byte long. This control entry is valid only for Functions 2, 3C and 3S. For Function 2: this control entry is required only if the input data file contains at least one input street intersection defined by a pair of streets that intersect at two distinct locations (see Section VII.2). In such input data records, the input compass direction field must contain a valid non-blank compass direction value, 'N', 'S', 'E' or 'W', which serves to designate which of the two intersections of the given pair of streets is to be processed. In other input data records, the input compass direction field should be blank. For Function 3C: this control entry is mandatory. The input data field that this control entry specifies contains the compass direction designating the side of the street. This field must contain a valid non-blank compass direction value, 'N', 'S', 'E' or 'W', in every input data record. For Function 3S: this control entry corresponds to the 'first input intersection', that is, the input intersection defined either by ONSTREET and CROSS1 or by STRTCODE and CRSCOD1. (The control entry COMPASS2 corresponds to the second input intersection.) The COMPASS control entry is required only if the input

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description
	data file contains at least one first input intersection that is defined by a pair of streets that intersect in two locations (see Section VII.2). In such input data records, the input data field that this control entry specifies must contain a valid compass direction value identifying which of the two locations is the intended first input intersection. In other input data records, this field should contain a blank.
COMPASS2=S	Specifies the position of the input compass direction field in the input data records that corresponds to the 'second input intersection', that is, the input intersection defined either by ONSTREET and CROSS2 or by STRTCODE and CRSCOD2. The COMPASS2 control entry is never coded with a length variable; GBAT always assumes an input compass direction field to be one byte long. The COMPASS2 control entry is valid only for Function 3S. It is required only if the input data file contains at least one second input intersection that is defined by a pair of streets that intersect at two distinct locations (see Section VII.2). In such input data records, the input data field that this control entry specifies must contain a valid compass direction value identifying which of the two locations is the intended second input intersection. In other input data records, this field should contain a blank.
CROSSBORO1=S,L	Specifies the starting position and length of the input borough code field corresponding to the input street field specified by the control entry CROSS1 or CRSCOD1. CROSSBORO1 is coded only if the input data records have separate borough code fields corresponding to each of the input street fields. Such separate input borough code fields enable GBAT to process input data files containing borough boundary locations that are defined by streets in different boroughs (see discussion of Geosupport's borough boundary processing feature in Section VII.7). CROSSBORO1 is valid for Functions 2, 3, 3C, 3S and D, and is optional for those functions. If CROSSBORO1 is not coded, then the field specified by BORO is used as the input borough code field for the input street field specified by CROSS1 or CRSCOD1. If CROSSBORO1 is coded, then whenever the field it specifies contains a blank, the contents of the field specified by BORO is used as the input borough code for the input street field specified by CROSS1 or CRSCOD1. If CROSSBORO1 is coded, and the input street field it applies to is specified by CRSCOD1 rather than CROSS1 (i.e., if that field contains street codes rather than street names), and the length of CRSCOD1 is specified as 4 or 6 (i.e., the input street code field is in one of the formats that contain their own borough code), then CROSSBORO1 is ignored, and the input borough code field it specifies is not used. Note: The input borough code field specified by CROSSBORO1 may contain user-defined, non-standard borough code values - see discussion at table entry for BRONX.
CROSSBORO2=S,L	Specifies the starting position and length of the input borough code field corresponding to the input street field specified by the control entry CROSS2 or CRSCOD2.

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description
	CROSSBORO2 is coded only if the input data records have separate borough code fields corresponding to each of the input street fields. Such separate input borough code fields enable GBAT to process input data files containing borough boundary locations that are defined by streets in different boroughs (see discussion of Geosupport's borough boundary processing feature in Section VII.7). CROSSBORO2 is valid for Functions 3, 3C, 3S and D, and is optional for those functions. If CROSSBORO2 is not coded, then the field specified by BORO is used as the input borough code field for the input street field specified by CROSS2 or CRSCOD2. If CROSSBORO2 is coded, then whenever the field it specifies contains a blank, the contents of the field specified by BORO is used as the input borough code for the input street field specified by CROSS2 or CRSCOD2. If CROSSBORO2 is coded, and the input street field it applies to is specified by CRSCOD2 rather than CROSS2 (i.e., if that field contains street codes rather than street names), and the length of CRSCOD2 is specified as 4 or 6 (i.e., the input street code field is in one of the formats that contain their own borough code), then CROSSBORO2 is ignored, and the input borough code field it specifies is not used. Note: The input borough code field specified by CROSSBORO2 may contain user-defined, non-standard borough code values - see discussion at table entry for BRONX.
CROSSSTNAMES=V	Specifies whether a list of street names of the cross streets or intersecting streets is to be included in the appended output data. Note: the CROSSSTNAMES feature incurs processing overhead, and should only be used when necessary. CROSSSTNAMES=YES specifies that the street names of the cross streets (RECTYPE=1, 1E, 3 or 3C) or intersecting streets (RECTYTPE=2) are to be appended. This causes a 320-byte block of data to be included in the appended data containing those street names, laid out as described in the Appendix 3 entry for the List of Street Names (see paragraph on List of Cross Street Names). CROSSSTNAMES=YES is valid only when GEOCODE=ALL and RECTYPE=1, 1E, 2, 3 or 3C have been specified. CROSSSTNAMES=NO specifies that the street names of cross streets or intersecting streets are not to be appended. This control entry is optional; if it is not coded, the default value is NO.
CROSS1, **CROSS2**	See ONSTREET.
CRSCOD1, **CRSCOD2**	See STRTCODE.
GEOCODE=V	Specifies whether GBAT will issue one-work-area or two-work-area calls (see

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description
	Section II.4); whether OUTFILE will be produced; and if so, what information GBAT will append to the user input records in forming the OUTFILE records (see Section IX.7). The valid variable values for this control entry are NO, YES, ALL and VAL. GEOCODE=NO specifies a one-work-area call. Only selected information from Work Area 1 is appended. For Function BL, the BBL is appended; for Function BN, the BIN is appended; for the other functions, normalized house numbers, normalized street names and street codes are appended. For a detailed layout of the appended information for GEOCODE=NO, see Table A9-4 for MSW format, and see Table A12-2 for COW format. OUTFILE is produced. GEOCODE=YES specifies a two-work-area call. Only a copy of Work Area 2 for the given function is appended. OUTFILE is produced. GEOCODE=ALL specifies a two-work-area call. Both the GEOCODE=NO information and the GEOCODE=YES information are appended. In addition, if CROSSSTNAMES=YES is specified, a list of street names of the cross streets or intersecting streets is also appended, in the form of a 320-byte block of data, between the GEOCODE=NO data and the GEOCODE=YES data. OUTFILE is produced. GEOCODE=VAL specifies a two-work-area call. OUTFILE is not produced. This control entry is optional. The default value depends on the function: it is NO for Functions 1, 1N, 2, 3, D, DG and DN, and it is YES for all other functions. The values YES, ALL and VAL are invalid for functions that can only be called using one work area (currently, Functions 1N, D, DG and DN).
HNI=V	Specifies whether the input house number fields specified by the HOUSENUM and HOUSENUM2 control entries are House Numbers in Internal format (HNIs) (see Section V.2) or are in character format, indicated by the variable values YES and NO respectively. The HNI control entry is optional only for MSW Functions 1, 1A, 1E, D, DG and DN, and is invalid for other functions and for the COW format. The default value depends on the function. For MSW Functions 1, 1A and 1E, NO is the default, and YES is also valid. For MSW Functions D, DG and DN, YES is the default and is the only valid value.

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

<u>Control Entry</u>	<u>Description</u>
HNS=V	Specifies whether the input house number field specified by the HOUSENUM and HOUSENUM2 control entries are House Numbers in Sort format (HNSs) (see Section V.2) or are in character format, indicated by the variable values YES and NO respectively. The HNS control entry is optional for COW Functions 1, 1A, 1E, D, DG and DN, and is invalid for other functions and for the MSW format. The default value depends on the function. For COW Functions 1, 1A and 1E, NO is the default, and YES is also valid. For COW Functions D, DG and DN, YES is the default and is the only valid value.
HOUSENUM=S or **HOUSENUM=S,L**	Specifies the starting position and length of an input house number field. This control entry is optional. It is valid for Functions 1, 1A, 1E, D, DG and DN. For Functions 1, 1A and 1E, if HOUSENUM is not coded, Geosupport assumes that the input street name field (specified by the control entry ONSTREET) contains a free-form address (see Section V.3). If HOUSENUM is coded, the input data field it specifies may contain either a House Number in Internal format (HNI - for MSW format only - see Section V.2), a House Number in Sort Format (HNS - for COW format only - see Section V.2), or a house number in character format. If it contains an HNI, then the control entry HNI=YES must be in effect (either by explicitly coding it or by default), and HOUSENUM's length variable must either be coded with the value '6' or not coded (in which case it defaults to '6' by virtue of HNI=YES). If it contains an HNS, then the control entry HNS=YES must be in effect (either by explicitly coding it or by default), and HOUSENUM's length variable must either be coded with the value '11' or not coded (in which case it defaults to '11' by virtue of HNS=YES). If HOUSENUM is not an HNI or an HNS, its length variable must be a number between 5 and 12. For Functions D, DG and DN, if HOUSENUM is coded, for MSW format, the input data field it specifies must contain an HNI, and the length variable must either be coded with the value '6' or not coded (in which case it defaults to '6'). For COW format, the input data field it specifies must contain an HNS, and the length variable must either be coded with the value '11' or not coded (in which case it defaults to '11').
HOUSENUM2=S or **HOUSENUM2=S,L**	Specifies the starting position and length of an input house number field containing an HNI for the MSW format, or an HNS for the COW format. HOUSENUM2 is valid for Functions D, DG and DN, for which it is optional. Those functions can accept two input HNI or HNS fields per call. Coding both HOUSENUM and HOUSENUM2 enables two input HNI or HNS fields to be processed through Functions D, DG or DN in a single GBAT pass. For the MSW format, when HOUSENUM2 is coded, the OUTFILE records include a corresponding 12-byte field (identified as HND-2 in Table A9-4) in the appended data for a House Number in Display format (HND), and the total length of the appended data is 120 bytes. When HOUSENUM2 is not coded using the MSW format, no corresponding HND-2 field is

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

<u>Control Entry</u>	<u>Description</u>
	included in the OUTFILE records, and the total length of the appended data is 108 bytes. For the COW format, the OUTFILE record length is always 128 bytes, and has space for two 16-byte output House Numbers in Display format (HNDs), each of which would either have data or be blank, dependent on the input..
LONGWA2=V	Specifies whether the Work Area 2 to be used to form the records written into the output file of accepted data (OUTFILE) is the regular WA2 or the long WA2 (see Section II.5). The valid variable values are YES and NO and are self-explanatory. Currently, the long WA2 option is only available for MSW format Functions 1, 1E, and 3, and for both MSW and COW formats for Functions 1A and BL; this control entry is invalid for all other functions. For the functions that have the long WA2 option, this control entry is invalid when GEOCODE=NO or VAL; it is optional when GEOCODE=YES or ALL, and the default value is NO.
LOT=S,L	Specifies the starting position and length of the input tax lot field. This control entry is valid only for Function BL. The Function BL user must specify either all three control entries BORO, BLOCK and LOT, or the control entry BBL. The length value of LOT must be explicitly coded as ‘4’. There is no default.
MANHATTAN	See BRONX.
MAXREJECTS=V	Specifies how many rejects (including warnings, if REJECTWARNINGS=YES has been specified) occurring at the beginning of the input data file, other than any records rejected for an invalid borough code, are to cause a ‘MAXREJECTS termination’, that is, would cause GBAT to terminate execution abnormally and exit with Condition Code 20 (see Section IX.3). The variable value must be either a positive integer specifying the number of such rejects that are to cause a MAXREJECTS termination, or the value NOMAX. If MAXREJECTS=NOMAX is coded, the entire input data file is processed, regardless of the number of rejects occurring at the beginning of the file. The MAXREJECTS control entry is optional, and the default value is MAXREJECTS=200.
ONSTREET=S,L **CROSS1=S,L** **CROSS2=S,L**	Specify the starting positions and lengths of input street name fields for Functions 1, 1A, 1E, 1N, 2, 3, 3C, 3S. (For functions 1, 1A, 1E, those fields may contain free-form addresses.) The appropriate combination of these control entries for the function being called must be coded, as follows:

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description		
	Functions	**Field(s) Specified**	**Control Entries Used to Specify These Fields**
	1, 1A, 1E, 1N	'On' Street	ONSTREET
	2	Two Intersecting Streets in Either Order	ONSTREET and CROSS1
	3, 3C, 3S	'On' Street and Two Cross Streets in Either Order	ONSTREET, CROSS1 and CROSS2 (Note: For Function 3S, input cross street fields are optional.)

Whenever input street name fields are specified by coding any of the three control entries ONSTREET, CROSS1 and CROSS2, an input borough code field (or fields) must also be specified, by coding the control entry BORO (and optionally CROSSBORO1 and CROSSBORO2, as appropriate).

Note: for Functions 1, 1A, 1E, 2, 3, 3C and 3S, but not function 1N, input street data may be provided either in the form of street name fields, specified using the control entries ONSTREET, CROSS1 and CROSS2, or alternatively, in the form of five-digit street code fields (see Section IV.8), specified using the control entries STRTCODE, CRSCOD1 and CRSCOD2. For those of the aforementioned functions that accept multiple input streets, either all of those input streets must take the form of street names or all must take the form of street codes; a mixture of names and codes is not permitted. For Function 1N, street name input fields are mandatory.

QUEENS See BRONX.

RECTYPE=V Specifies the Geosupport function to be executed. This control entry is mandatory. The valid variable values are the valid Geosupport function codes. As of this writing, these are 1, 1A, 1E, 1N, 2, 3, 3C, 3S, BL, BN, D, DG, DN.

REJECTWARNINGS=V

Specifies whether input data records that result in warnings are to be treated as accepted records or as rejects (see discussion of REJECTWARNINGS in Section IX.7). The valid variable values are YES and NO. This control entry is optional, and the default value is NO.

If REJECTWARNINGS=YES is coded, records resulting in warnings (GRC='01') are treated as rejects; that is, they are written into ERRFILE rather than OUTFILE, they are counted as rejects in the report of run statistics, and they are considered to be rejects for the purpose of determining whether a MAXREJECTS termination is to be triggered.

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description
	If REJECTWARNINGS=NO is coded or is in effect by default, warnings are treated as accepted records; that is, they are written into OUTFILE rather than ERRFILE, they are counted as accepted records in the report of run statistics, and they are considered to be accepted records for the purpose of determining whether a MAXREJECTS termination is to be triggered.
ROADBED=V	Specifies whether the output of a multi-Roadbed street should contain Roadbed information or information based the center line of the street. This control entry is valid only for Functions 1 and 1E, for which it is optional. The variable values are YES and NO and are self-explanatory. If this control entry is not coded, the default value is NO.
SNL=V	Specifies a value for the Street Name Normalization Length Limit (SNL) parameter (see Section III.2). The variable value must be a number between 4 and 32, inclusive. This control entry is valid only for functions that return normalized street names, for which it is optional. The default value is 32.
STATEN	See BRONX.
STRTCODE=S,L **CRSCOD1=S,L** **CRSCOD2=S,L**	Specify the starting positions and lengths of input five-digit street code fields for Functions 1, 1A, 1E, 2, 3, 3C, 3S and D. The appropriate combination of these control entries for the function being called must be coded, as follows:

Functions	**Street Input Fields Required**	**Control Entries Used to Specify These Fields**
1, 1A, 1E	'On' Street	STRTCODE
2	Two Intersecting Streets in Either Order	STRTCODE and CRSCOD1
3, 3C, 3S	'On' Street and Two Cross Streets in Either Order	STRTCODE, CRSCOD1 and CRSCOD2 (Note: For Function 3S, input cross street fields are optional.)
D	Up to Three 5-Digit Street Codes	STRTCODE, CRSCOD1 if necessary, CRSCOD2 if necessary

Each of the input street code fields specified by STRTCODE, CRSCOD1 and CRSCOD2 must contain a five-digit street code in one of four formats: P5SC (which has a length of 3 bytes, valid only with MSW format), PB5SC (length = 4, valid only with MSW format), 5SC (length = 5) or B5SC (length = 6). For Functions 2, 3, 3C, 3S and D, which can have multiple input five-digit street code fields, it is allowable

Table A9-1: GBAT Control Entry Descriptions by Keyword (continued)

Control Entry	Description
	for those fields to have different formats; for example, for MSW Function 3, it is permissible for the input ‘on’ street field to contain a PB5SC while one input cross street field contains a 5SC and the other input cross street field contains a B5SC. The user must code the length variable value in each of these control entries so that it accords with the street code format of the corresponding input street code field. GBAT uses that length value to determine which five-digit street code format to expect in that input field. If any input street code fields are in the form of P5SCs or 5SCs, which do not contain their own borough code sub-field, the input data file must also have a separate input borough code field or fields, which must be specified by coding the control entry BORO, and if needed, the control entries CROSSBORO1 and CROSSBORO2. Note: for Functions 1, 1A, 1E, 2, 3, 3C and 3S, but not Function D, input street data may be provided either in the form of five-digit street code fields, specified using the control entries STRTCODE, CRSCOD1 and CRSCOD2, or alternatively, in the form of street name fields, specified using the control entries ONSTREET, CROSS1 and CROSS2. For those of the aforementioned functions that accept multiple input streets, either all of those input streets must take the form of street codes or all must take the form of street names; a mixture of codes and names is not permitted. For Function D, input street data must be in the form of street codes.
TITLE=V	Specifies a title to appear on the top of the SYSPRINT output report of summary run statistics. A valid variable value is any character string of up to 73 bytes ending in a semicolon. This control entry is optional. If it is not coded, the report is generated without a title. If it is coded, it is mandatory to terminate the title character string with a semicolon, which does not appear in the actual report.
VSAM=V	Specifies whether the input data file is a VSAM file or a sequential file. The valid variable values are YES and NO, specifying that the file is a VSAM file or a sequential file, respectively. This control entry is optional, and NO is the default value. If NO is specified or is in effect by default, then in the JCL, the DD statement for the input data file must contain the DDname INFILE. If YES is specified, the DDname must be coded as INVSAM.
WORKAREA=V	Specifies whether the work areas should be in MSW or COW format. The WORKAREA control entry is valid for all functions. The valid variable values are COW (Character Only Work Area) and MSW (Mainframe Specific Work Area), and are self-explanatory. If this control entry is not coded, the default value is MSW.
1ABLVERSION=V	Specifies that standard processing is to be performed for Functions 1A and BL (see Section VI.8). The 1ABLVERSION control entry is valid only for Functions 1A and BL, and is required for the MSW format. The only valid variable value for this control entry is STANDARD or S, and is self-explanatory. Note: Legacy is no longer supported.

Table A9-2: Summary of GBAT Control Entries by Keyword

This table lists all of the control entries alphabetically by keyword, indicates their coding formats, their allowable and default variable values, and the Geosupport functions for which each control entry or combination of control entry and variable value is valid. Control entry variables are indicated using ‘S’ and ‘L’ to represent the starting position and length of a field in the input data records, respectively, and ‘V’ to represent variables of other types.

Control Entry	Valid Variable Values	Default	Functions
ALIASES=V	NO, YES, VAL	NO	1, 1A, 1E, 1N, 2, 3, 3C, 3S
BBL=S,10	1 ≤ S ≤ (LRECL-1)-10	None	BL
BIN=S or BIN=S,7	1 ≤ S≤ (LRECL+1)-7	L=7	BN
BLOCK=S or BLOCK=S,5	1 ≤ S ≤ (LRECL+1)-5	L=5	BL
BORO=S,L	1 ≤ S ≤ (LRECL+1)-L 1≤ L ≤ 12	None	All but BN, DG, DN
BRONX=V	Any character string that fits BORO	2	All but BN, DG, DN
BROOKLYN=V	Any character string that fits BORO	3	All but BN, DG, DN
B7SC1=S or B7SC1=S,8	1 ≤ S ≤ (LRECL+1)-8	L=8	DG
B7SC2=S or B7SC2=S,8	1 ≤ S ≤ (LRECL+1)-8	L=8	DG
B7SC3=S or B7SC3=S,8	1 ≤ S ≤ (LRECL+1)-8	L=8	DG
B10SC1=S or B10SC1=S,11	1 ≤ S ≤ (LRECL+1)-11	L=11	DN
B10SC2=S or B10SC2=S,11	1 ≤ S ≤ (LRECL+1)-11	L=11	DN
B10SC3=S or B10SC3=S,11	1 ≤ S ≤ (LRECL+1)-11	L=11	DN
COMPACT=V	YES, NO	NO	All but BL, BN
COMPASS=S	1 ≤ S ≤ LRECL	None	2, 3C, 3S
COMPASS2=S	1 ≤ S ≤ LRECL	None	3S
CROSSBORO1=S,L	1 ≤ S ≤ (LRECL+1)-L 1≤ L ≤ 12	None	2, 3, 3C, 3S, D
CROSSBORO2=S,L	1 ≤ S ≤ (LRECL+1)-L 1≤ L ≤ 12	None	3, 3C, 3S, D
CROSSSTNAMES=V	YES, NO (YES is valid only for GEOCODE=ALL)	NO	1, 1E, 2, 3, 3C

Table A9-2: Summary of GBAT Control Entries by Keyword (continued)

Control Entry	Valid Variable Values	Default	Functions
CROSS1=S,L	$1 \le S \le (LRECL+1)-L$ $4 \le L \le 32$	None	2, 3, 3C, 3S
CROSS2=S,L	$1 \le S \le (LRECL+1)-L$ $4 \le L \le 32$	None	3, 3C, 3S
CRSCOD1=S,L	$1 \le S \le (LRECL+1)-L$ L=3 if field contains P5SC (MSW) L=4 if field contains PB5SC (MSW) L=5 if field contains 5SC L=6 if field contains B5SC	None	2, 3, 3C, 3S, D
CRSCOD2=S,L	$1 \le S \le (LRECL+1)-L$ L=3 if field contains P5SC (MSW) L=4 if field contains PB5SC (MSW) L=5 if field contains 5SC L=6 if field contains B5SC	None	3, 3C, 3S, D
GEOCODE=V	NO, YES, ALL, VAL	NO	1, 1N, 2, 3, D, DG, DN
	(YES and ALL are invalid for Functions 1N, D, DG, DN)	YES	1A, 1E, 3C, 3S, BL, BN
HNI=V	YES, NO (MSW format only)	YES	D, DG, DN
	(NO is invalid for MSW Fns D, DG, DN)	NO	1, 1A, 1E
HNS=V	YES, NO (COW format only)	YES	D, DG, DN
	(NO is invalid for COW Fns D, DG, DN)	NO	1, 1A, 1E
HOUSENUM=S or HOUSENUM=S,L	$1 \le S \le (LRECL+1)-L$ $5 \le L \le 12$ if field contains house number in character format L= 6 if field contains HNI L=11 if field contains HNS	L=6 when HNI=YES; L=11 when HNS=YES else no length default	1, 1A, 1E, D, DG, DN
HOUSENUM2=S or HOUSENUM2=S,L	$1 \le S \le (LRECL+1)-L$ L= 6 if field contains HNI L=11 if field contains HNS	L=6 when HNI=YES; L=11 when HNS=YES	D, DG, DN

Table A9-2: Summary of GBAT Control Entries by Keyword (continued)

Control Entry	Valid Variable Values	Default	Functions
LONGWA2=V	YES, NO	NO	MSW and COW: 1A, BL MSW only: 1, 1E, 3
LOT=S,4	1 ≤ S ≤ (LRECL+1)-4	None	BL
MANHATTAN=V	Any character string that fits BORO	1	All but BN, DG, DN
MAXREJECTS=V	Any positive integer or NOMAX	200	All
ONSTREET=S,L	1 ≤ S≤ (LRECL+1)-L 4 ≤ L ≤ 32	None	1,1A,1E,1N, 2,3,3C, 3S
QUEENS=V	Any character string that fits BORO	4	All but BN, DG, DN
RECTYPE=V	1, 1A, 1E, 1N, 2, 3, 3C, 3S, BL, BN, D, DG, DN	None	All
REJECTWARNINGS=V	YES, NO	NO	All
ROADBED	YES, NO	NO	1, 1E
SNL=V	4 ≤ V ≤ 32	32	All but BL, BN
STATEN=V	Any character string that fits BORO	5	All but BN, DG, DN
STRTCODE=S,L	1 ≤ S ≤ (LRECL+1)-L L=3 if contains P5SC (MSW) L=4 if contains PB5SC (MSW) L=5 if contains 5SC L=6 if contains B5SC	None	1, 1A, 1E, 2, 3, 3C, 3S, D
TITLE=V	Any character string of up to 73 bytes ending in a semicolon	No title	All
VSAM=V	YES, NO	NO	All
WORKAREA	COW, MSW	MSW	All
1ABLVERSION=V	STANDARD, S	None	1A, BL

Table A9-3: Summary of GBAT Control Entry Usage by Function

This table lists, by Geosupport function, which control entries are allowable and which of those are mandatory. Control entries are represented in this table by their keywords. Some combinations of control entries are mandatory or prohibited; such conditions are indicated in this table by using the logical connectors "and", "or" (inclusive or) and "xor" (exclusive or) and by using underlining, as follows:

- A table entry of the form "A and B" signifies that if either A or B is coded, then both must be coded. Similarly, "A and B and C" signifies that if any of A, B or C is coded, then all three must be coded.
- A table entry of the form "A or B" signifies that A may be coded without B, B may be coded without A, and A and B may both be coded. Similarly, "A or B or C" signifies that any combination of these three items may be coded.
- A table entry of the form "A xor B" signifies that if either A or B is coded, then the other one must not be coded.
- If a table entry is <u>underlined,</u> that control entry or combination of control entries is mandatory for the given function. All table entries not underlined are optional.
- Square brackets ("[.....]") are sometimes used for logical grouping to increase clarity.

Thus, a table entry of the form "<u>A or B</u>" signifies that it is mandatory to code A or B; that is, it is mandatory to code at least one of A and B and it is permissible to code both A and B. "<u>A xor B</u>" signifies that it is mandatory to code A xor B; that is, it is mandatory to code either A or B but prohibited (because of the exclusive or) to code both A and B. A table entry of the form "<u>[A and B] xor [C and D]</u>" signifies that it is mandatory to code either both A and B or both C and D but prohibited to code all four of them.

Function	**Control Entries**
1	ALIASES, <u>BORO</u>, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, CROSSSTNAMES, GEOCODE, HNI xor HNS (see Note 1), HOUSENUM (see Note 2), LONGWA2 xor WORKAREA=COW, MAXREJECTS, <u>ONSTREET xor STRTCODE</u>, <u>RECTYPE</u>, REJECTWARNINGS, ROADBED, SNL, TITLE, WORKAREA, VSAM <u>Note 1</u>: HNI is a valid entry only if WORKAREA defaults to MSW or is set to MSW. HNS is a valid entry only if WORKAREA=COW. <u>Note 2</u>: For Functions 1, 1A and 1E, HOUSENUM is optional in the sense that coding it is either mandatory or prohibited, depending, respectively, on whether the input data file contains free-form addresses (in which a single field contains the house number followed by the street name in non-fixed positions; see Section V.3) or parsed-form addresses (in which the house number and street name are in separate fields). When HOUSENUM is not coded, the input street must be in the form of street names rather than street codes, the input street name field must be specified by the control entry ONSTREET, and in every input data record, that field must contain either a free-form address or a Non-Addressable Place name (NAP). When HOUSENUM is coded, and the input street is in the form of street names, GBAT assumes that the input street name field contains street names and NAPs only, not

Table A9-3: Summary of GBAT Control Entry Usage by Function (continued)

Function	Control Entries
	free-form addresses.
1A	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, GEOCODE, HNI xor HNS (see Note 1 following Function 1 entry), HOUSENUM (see Note 2 following Function 1 entry), LONGWA2, MAXREJECTS, ONSTREET xor STRTCODE, RECTYPE, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM, 1ABLVERSION
1E	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, CROSSSTNAMES, GEOCODE, HNI xor HNS (see Note 1 following Function 1 entry), HOUSENUM (see Note 2 following Function 1 entry), LONGWA2 xor WORKAREA=COW, MAXREJECTS, ONSTREET xor STRTCODE, RECTYPE, REJECTWARNINGS, ROADBED, SNL, TITLE, WORKAREA, VSAM
1N	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, GEOCODE, MAXREJECTS, ONSTREET, RECTYPE, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
2	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, COMPASS, CROSSBORO1, CROSSSTNAMES, GEOCODE, MAXREJECTS, [ONSTREET and CROSS1] xor [STRTCODE and CRSCOD1], RECTYPE, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
3	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, CROSSBORO1, CROSSBORO2, CROSSSTNAMES, GEOCODE, LONGWA2 xor WORKAREA=COW, MAXREJECTS, [ONSTREET and CROSS1 and CROSS2] xor [STRTCODE and CRSCOD1 and CRSCOD2], RECTYPE, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
3C	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, COMPASS, CROSSBORO1, CROSSBORO2, CROSSSTNAMES, GEOCODE, MAXREJECTS, [ONSTREET and CROSS1 and CROSS2] xor [STRTCODE and CRSCOD1 and CRSCOD2], RECTYPE, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
3S	ALIASES, BORO, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, COMPASS, COMPASS2, CROSSBORO1, CROSSBORO2, [CROSS1 and CROSS2] xor STRTCODE, [CRSCOD1 and CRSCOD2] xor ONSTREET, GEOCODE, MAXREJECTS, ONSTREET xor STRTCODE, RECTYPE, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
BL	BORO and BLOCK and LOT xor BBL, BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, LONGWA2, MAXREJECTS, RECTYPE, REJECTWARNINGS, TITLE, VSAM, WORKAREA, 1ABLVERSION

Table A9-3: Summary of GBAT Control Entry Usage by Function (continued)

Function	**Control Entries**
BN	<u>BIN</u>, GEOCODE, MAXREJECTS, <u>RECTYPE</u>, REJECTWARNINGS, TITLE, WOKAREA, VSAM
D	BRONX and BROOKLYN and MANHATTAN and QUEENS and STATEN, COMPACT, CROSSBORO1, CROSSBORO2, GEOCODE, HNI xor HNS, <u>HOUSENUM or HOUSENUM2 or [BORO and STRTCODE] or [BORO and STRTCODE and CRSCOD1] or [BORO and STRTCODE and CRSCOD1 and CRSCOD2]</u>, MAXREJECTS, <u>RECTYPE</u>, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
DG	COMPACT, GEOCODE, HNI xor HNS, <u>HOUSENUM or HOUSENUM2 or B7SC1 or [B7SC1 and B7SC2] or [B7SC1 and B7SC2 and B7SC3]</u>, MAXREJECTS, <u>RECTYPE</u>, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM
DN	COMPACT, GEOCODE, HNI xor HNS, <u>HOUSENUM or HOUSENUM2 or B10SC1 or [B10SC1 and B10SC2] or [B10SC1 and B10SC2 and B10SC3]</u>, MAXREJECTS, <u>RECTYPE</u>, REJECTWARNINGS, SNL, TITLE, WORKAREA, VSAM

Table A9-4: MSW Appended Items for GEOCODE=NO

NOTE: For COW Appended Items for GEOCODE=NO see TABLE 12.2

This table contains, by function or combination of function and option, a layout of the data that GBAT appends to the input data record to form the OUTFILE record for the MSW format when GEOCODE=NO has been specified.

Function(s)	Option	Appended Items	Length
1, 1A, 1E		HND HNHPD Normalized Street Name 10SC	12 8 32 10
		Total Length:	62
1N		Normalized Street Name 10SC	32 10
		Total Length:	42
2		Normalized Street Name-1 10SC-1 Normalized Street Name-2 10SC-2	32 10 32 10
		Total Length:	84
3, 3C, 3S		Normalized Street Name-1 10SC-1 Normalized Street Name-2 10SC-2 Normalized Street Name-3 10SC-3	32 10 32 10 32 10
		Total Length:	126
BL	1ABLVERSION=STANDARD	BBL (Standard Format): Borough Code Tax Block Tax Lot	 1 5 4
		Total Length:	10
	1ABLVERSION=LEGACY	No longer supported	
BN		BIN Filler	7 3
		Total Length:	10
D, DG, DN	HOUSENUM2 control entry not coded	HND Normalized Street Name-1 Normalized Street Name-2 Normalized Street Name-3	12 32 32 32
		Total Length:	108

Table A9-4: MSW Appended Items for GEOCODE=NO (continued)

Function(s)	Option	Appended Items	Length
D, DG,DN (cont'd)	HOUSENUM2 control entry coded	HND-1 Normalized Street Name-1 Normalized Street Name-2 Normalized Street Name-3 HND-2	12 32 32 32 12
		Total Length:	120

Table A9-5: MSW Format- Length of GBAT-Appended Data

This table lists, by function and GEOCODE value, the length in bytes of the data that GBAT appends to an input data record that has been accepted by Geosupport to form the corresponding OUTFILE record. The LRECL value that the user must specify in the OUTFILE DD statement in the JCL is computed by adding the length of the appended data as indicated in this table to the LRECL of the input data file.

Note: For Functions 1, 1E, 2, 3 and 3C, when GEOCODE=ALL and CROSSSTNAMES=YES, the appended data consist of the concatenation of the GEOCODE=NO data, followed by a 320-byte block of data containing cross street names, followed by the GEOCODE=YES data. For further information about the layout of the appended CROSSSTNAMES data, refer to the Appendix 3 entry for the List of Street Names (see paragraph on List of Cross Street Names).

Functions	Options	GEOCODE Value		
		NO	YES	ALL
1, 1E		62		
	LONGWA2=NO, CROSSSTNAMES=NO		200	262
	LONGWA2=YES, CROSSSTNAMES=NO		300	362
	LONGWA2=NO, CROSSSTNAMES=YES			582
	LONGWA2=YES, CROSSSTNAMES=YES			682
1A	1ABLVERSION=STANDARD	62		
	1ABLVERSION=STANDARD, LONGWA2=NO		939	1001
	1ABLVERSION=STANDARD,LONGWA2=YES		17683	17745
1N		42	Invalid	Invalid
2	CROSSSTNAMES=NO	84	200	284
	CROSSSTNAMES=YES			604
3		126		
	LONGWA2=NO, CROSSSTNAMES=NO		200	326
	LONGWA2=YES, CROSSSTNAMES=NO		300	426
	LONGWA2=NO, CROSSSTNAMES=YES			646
	LONGWA2=YES, CROSSSTNAMES=YES			746
3C	CROSSSTNAMES=NO	126	200	326
	CROSSSTNAMES=YES			646
3S		126	4224	4350

Table A9-5: MSW Format Length of GBAT-Appended Data (continued)

Functions	Options	GEOCODE Value		
		NO	YES	ALL
BL	1ABLVERSION=STANDARD	10		
	1ABLVERSION=STANDARD, LONGWA2=NO		939	949
	1ABLVERSION=STANDARD, LONGWA2=YES		17683	17693
BN		10	939	949
D, DG, DN	HOUSENUM2 not coded	108	Invalid	Invalid
	HOUSENUM2 coded	120	Invalid	Invalid

APPENDIX 10: SAMPLE GBAT JOBS

This appendix contains printouts of two sample GBAT jobs, referred to as Sample Job 1 and Sample Job 2. Sample Job 1 executes Function 1A. Sample Job 2 executes Function 2.

For each sample job, this appendix contains a description of the control file, followed by listings of the job-stream input and the job output. The job-stream input listing contains the JCL, the in-stream control file, the in-stream data input file, and for Sample Job 2 only, the in-stream ALIASES input file. An ALIASES file is not used in Sample Job 1. The job output listing contains the system job-stream output, the GBAT output report of messages and run statistics, and the output file of GBAT rejects.

GSS developed and ran the sample jobs on the DoITT/Computer Service Center mainframe. Some variations from the JCL shown herein may be necessary for users running on other computers. In addition, the JCL shown has been modified to remove account-specific references.

Please note that the GBAT samples are not guaranteed to run exactly as shown in this appendix. The samples are here as an aid in developing GBAT runs.

Note: The GBAT samples are MSW format samples. To run using the **COW format**, just add the GBAT control entry WORKAREA=COW to the in-stream control files.

SAMPLE GBAT JOB #1

SAMPLE GBAT JOB 1: DESCRIPTION

The control file for Sample GBAT Job 1 is as follows:

```
BORO=9,1   RECTYPE=1A    ONSTREET=15,20    1ABLVERSION=S
TITLE=THIS IS GBAT CONTROL FILE EXAMPLE 1;
```

In this example, the user has chosen to code several control entries in a single control record, followed by a second control record containing a heading for the SYSPRINT output file. In the first control record, the order in which the control entries are coded, their precise positioning within the control record and the amount of spacing between them are immaterial.

The control file in this example contains the following control entries. (See Table A9-1 for Control Entry Descriptions)

- BORO specifies that the input borough code field is in position 9 of the INFILE records and is one byte long.
- RECTYPE specifies Function 1A.
- ONSTREET specifies that the input street name field starts in position 15 of the INFILE records and is 20 bytes long.
- 1ABLVERSION specifies that the standard version of Function 1A is to be executed (and therefore that the appended Work Area 2 will be in the standard rather than the legacy format).
- TITLE specifies a title for the SYSPRINT output report. Notice that the text of the title is terminated with a semicolon character, as required. (The semicolon does not appear in the actual report.)

The user has chosen not to code the following control entries, the default values for which are therefore in effect: (See Table A9-2 for Control Entry default values.)

- Since the control entry VSAM has not been coded, GBAT will assume that the user input data file is a sequential file. Consequently, GBAT will access the input data file via the DDname INFILE, and the corresponding DD statement in the JCL must be coded accordingly.
- Since the control entry GEOCODE has not been coded, and Function 1A is being executed, the default value of YES is in effect. This causes GBAT to issue a two-work-area call and to append Work Area 2 for Function 1A to the successfully processed INFILE records in forming the OUTFILE records.
- Since the control entry ALIASES has not been coded, the default value of NO is in effect. Therefore, GBAT will not use temporary user-defined aliases when processing input street names; if an ALIASES DD statement has been included in the JCL, it will be ignored.
- Since the control entries MANHATTAN, BRONX etc. have not been coded, GBAT will assume that the input borough code field contains the default borough code values, which are the standard

Geosupport borough codes (‘1’ for Manhattan, ‘2’ for the Bronx, etc.).

- Since HNI (or HNS for COW) has not been coded, GBAT will assume that input house numbers are not necessarily normalized and are in display format rather than in the HNI (or HNS for COW) format.

- Since HOUSENUM has not been coded, GBAT will assume that the input street name field specified by ONSTREET contains a free-form address (a house number followed by a street name). Note that since in this GBAT run input addresses are free-form, partial street names (see Section III.4) will be rejected.

- Since COMPACT has not been coded, GBAT will return normalized street names in a format suitable for sorting, rather than in the compact format.

- Since REJECTWARNINGS has not been coded, the default value of NO is in effect, so warnings will be treated as successfully processed records: they will be written to OUTFILE, they will be counted as successfully processed records in the SYSPRINT report, and they will not be counted as rejects towards the MAXREJECTS termination limit.

- Since MAXREJECTS has not been coded, the default value of ‘200’ is in effect, so that GBAT will terminate with an MVS Return Code of ‘20’ if the first 200 INFILE records all result in rejects for any reason other than an invalid borough code.

- Since the SNL control entry has not been coded, GBAT will assume the default value of SNL=32 when normalizing street names.

- Since LONGWA2=YES has not been coded, GBAT will return the regular WA2 for Function 1A.

SAMPLE GBAT JOB 1: JOB-STREAM INPUT

```
//EXAMPLE1 JOB YOUR-JOB-CARD-INFORMATION
//*
//*
//*******************************************************************
//*************     THIS JOB IS GBAT MSW EXAMPLE 1     ***************
//*******************************************************************
//*******************************************************************
//*********   THIS STEP INVOKES THE STANDARD CATALOGUED      ********
//*********   PROCEDURE FOR GBAT EXECUTION, CALLED GBAT2     ********
//*******************************************************************
//S1 EXEC GBAT2
//*******************************************************************
//*********   CARDIN IS THE USER-PROVIDED CONTROL FILE       ********
//*******************************************************************
//CARDIN DD *
BORO=9,1  RECTYPE=1A ONSTREET=15,20 1ABLVERSION=S
TITLE=THIS IS GBAT CONTROL FILE MSW EXAMPLE 1;
/*
//*******************************************************************
//*********   INFILE IS THE USER-PROVIDED INPUT DATA.           ******
//*********   IN THIS EXAMPLE, IT IS PROVIDED AS INSTREAM DATA.******
//*******************************************************************
//INFILE DD *
        1     100 GARAGE CENTRE ST
        1     22  READE ST
        1     36  READE ST
        1     60  READE             ** PARTIAL STREET NAMES NOT ALLOWED
        1     12  ELK               ** IN FREE-FORM ADDRESSES
        1     12  ELK ST
        1     310   BWY
        1     99 W 3 ST
        2     709 E 165 ST
        2     187C  EDGEWATER PK
        4     229-16 87 AVE
        2     1475  LONGFELLOW AV
        1     2053 ADAM POWELL BL
        1     310   1 AVE
/*
//*******************************************************************
//*********   OUTFILE IS THE OUTPUT FILE OF SUCCESSFULLY     ********
//*********   PROCESSED INFILE RECORDS.                      ********
//*******************************************************************
//OUTFILE DD DSN=&&OUT1A,DISP=(NEW,PASS),
//        UNIT=SYSDA,SPACE=(TRK,(80,20),RLSE),
//        DCB=(RECFM=FB,LRECL=1019)
//*******************************************************************
//*********   ERRFILE IS THE OUTPUT FILE OF REJECTED          ********
//*********   INFILE RECORDS.                                ********
//*******************************************************************
//ERRFILE DD SYSOUT=A,DCB=(RECFM=FB,LRECL=84)
//*******************************************************************
//*********   AS OF GEOSUPPORT VERSION 10.0,                 ********
//*********   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G.  ********
//*********   GRID, PAD, TABFILE ETC) ARE NO LONGER NEEDED   ********
//*********   AND ARE IGNORED.  GEOSUPPORT IS TAILORED TO    ********
//*********   USE STANDARD GEOSUPPORT DATA SET NAMES.        ********
//*********   TO USE NON-STANDARD FILES, PLEASE SEE YOUR     ********
//*********   SYSTEMS PROGRAMMER.                            ********
//*******************************************************************
//
```

SAMPLE GBAT JOB 1: OUTPUT

```
                    J E S 2  J O B  L O G  --  S Y S T E M  M V S P  --  N O D E  C S C B A T C H

10.29.19 JOB17476 ---- FRIDAY,    DD MMMM YYYY ----
10.29.19 JOB17476  IRR010I  USERID YOURUID  IS ASSIGNED TO THIS JOB.
10.29.19 JOB17476  ICH70001I YOURUID  LAST ACCESS AT 10:27:49 ON FRIDAY, MMMM DD, YYYY
10.29.19 JOB17476  $HASP373 EXAMPLE1 STARTED - INIT 84   - CLASS X - SYS MVSP
10.29.19 JOB17476  IEF403I EXAMPLE1 - STARTED - TIME=10.29.19
10.29.19 JOB17476  +GBI SUCCESSFULLY LOADED GBIDRV
10.29.20 JOB17476  +GBIDRV (VERSION VV.V) INVOKED
10.29.20 JOB17476  +GEO (VERSION VV.V) INVOKED
10.29.20 JOB17476  +snd NNN  OPENED SUCCESSFULLY
10.29.20 JOB17476  +PAD NNN  'B030.GEO.COW.BLDGS.CITY' OPENED SUCCESSFULLY
10.29.20 JOB17476  -                                                --TIMINGS (MINS.)--            ----PAGING COUNTS---
10.29.20 JOB17476  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK   SERV  PG  PAGE  SWAP   VIO SWAPS
10.29.20 JOB17476  -EXAMPLE1 S1       GBAT2       00    787    142    .00    .00     .0   1198   0     0     0     0     0
10.29.20 JOB17476  IEF404I EXAMPLE1 - ENDED - TIME=10.29.20
10.29.20 JOB17476  -EXAMPLE1 ENDED.  NAME-YOURUID             TOTAL TCB CPU TIME=   .00  TOTAL ELAPSED TIME=    .0
10.29.20 JOB17476  $HASP395 EXAMPLE1 ENDED

------ JES2 JOB STATISTICS ------


  DD MMMM YYYY JOB EXECUTION DATE


          60 CARDS READ


         208 SYSOUT PRINT RECORDS


           0 SYSOUT PUNCH RECORDS


          15 SYSOUT SPOOL KBYTES


        0.01 MINUTES EXECUTION TIME
```

```
1 //EXAMPLE1 JOB YOUR-JOB-CARD-INFORMATION
  //*
  //*
  //*********************************************************************
  //************     THIS JOB IS GBAT MSW EXAMPLE 1     ***************
  //*********************************************************************
  //*********************************************************************
  //*********   THIS STEP INVOKES THE STANDARD CATALOGUED      ********
  //*********   PROCEDURE FOR GBAT EXECUTION, CALLED GBAT2     ********
  //*********************************************************************
3 //S1 EXEC GBAT2
4 XXGBAT2 PROC                                                           00000100
  XX**                      /* IN CSC.TEST.PROCLIB */                    00000200
  XX**                      /* MODIFIED 06/30/06  BY MEB   */            00000315
  XX**                      /* ADDED SUPPORT.PDSE.LOADLIB  */            00000415
  XX**                      /* REMOVED DD CARDS            */            00000515
  XX**                      /* MODIFIED 05/11/06  BY MEB   */            00000615
  XX**                      /* ADDED GRID1R FILE           */            00000715
  XX**                      /* MODIFIED 07/25/05  BY MEB   */            00000815
  XX**                      /* PEDFILE BECOMES DUMMY FILE  */            00000915
  XX**                      /* MODIFIED 03/26/02  BY MEB   */            00001015
  XX**                                                                   00001115
  XX**** WARNING: DO NOT OVERRIDE THE REGION PARAMETER  **************    00001215
  XX**                                                                   00001315
5 XXGBAT2 EXEC PGM=GBATIO2,REGION=9M,PARM='ISASIZE(40K)'                 00001415
  XX*                                                                    00001516
  XX*******************************************************************/  00001616
  XX*                                                               */   00001716
  XX*   AS OF GEOSUPPORT VERSION 10.0,                              */   00001816
  XX*   THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP   */   00001916
  XX*   MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:      */   00002016
  XX*               A030.GEO.SUPPORT.PDSE.LOADLIB                  */   00002116
  XX*               A030.GEO.SUPPORT.LOADLIB                       */   00002216
  XX*                                                               */   00002316
  XX*******************************************************************/  00002416
  XX*                                                                    00002516
  XX*                                                                    00002616
6 XXSTEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR                00002716
7 XX        DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR                     00002816
  XX*                                                                    00002916
  XX*                                                                    00003016
  XX*******************************************************************/  00003116
  XX*                                                               */   00003216
  XX*   AS OF GEOSUPPORT VERSION 10.0,                              */   00003316
  XX*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD    */   00003416
  XX*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED.  GEOSUPPORT     */   00003516
  XX*   IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.     */   00003616
  XX*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.    */   00003716
```

```
   XX*                                                                 */     00003816
   XX*******************************************************************/     00003916
   XX*                                                                 */     00004016
   XX*                                                                 */     00004116
 8 XXSYSPRINT DD SYSOUT=A,DCB=(LRECL=133,RECFM=FBA,BLKSIZE=1330)              00004216
   XX** SYSPRINT FILE CONTAINS RUN STATISTICS AND MESSAGES                    00004316
 9 XXSYSTERM  DD SYSOUT=A,DCB=(LRECL=133,RECFM=FBA,BLKSIZE=1330)              00004416
   XX** SYSTERM FILE CONTAINS SYSTEM WARNINGS AND ERRORS                      00004516
10 //CARDIN DD *
   X/CARDIN DD DDNAME=CARDIN                                                  00004616
   XX** CARDIN IS THE FILE OF GBAT CONTROL RECORDS                            00004716
11 //INFILE DD *
   X/INFILE DD DDNAME=INFILE                                                  00004816
   XX** INFILE CONTAINS THE USERS DATA INPUT RECORDS                          00004916
12 //OUTFILE DD DSN=&&OUT1A,DISP=(NEW,PASS),
   //        UNIT=SYSDA,SPACE=(TRK,(80,20),RLSE),
   //        DCB=(RECFM=FB,LRECL=1019)
   X/OUTFILE DD DDNAME=OUTFILE                                                00005016
   XX** OUTFILE CONTAINS THE VALID OUTPUT RECORDS                             00005116
13 //ERRFILE DD SYSOUT=A,DCB=(RECFM=FB,LRECL=84)
   X/ERRFILE DD DDNAME=ERRFILE                                                00005216
   XX** ERRFILE CONTAINS THE REJECTS                                          00005316
14 XXALIASES  DD DUMMY                                                        00005416
   XX** ALIASES IS THE OPTIONAL INPUT FILE OF USER-DEFINED ST NAME ALIASES 00006015
   //*********************************************************************
   //*********   CARDIN IS THE USER-PROVIDED CONTROL FILE       ********
   //*********************************************************************
   //*********************************************************************
   //*********   INFILE IS THE USER-PROVIDED INPUT DATA.          ******
   //*********   IN THIS EXAMPLE, IT IS PROVIDED AS INSTREAM DATA.******
   //*********************************************************************
   //*********************************************************************
   //*********   OUTFILE IS THE OUTPUT FILE OF SUCCESSFULLY     ********
   //*********   PROCESSED INFILE RECORDS.                      ********
   //*********************************************************************
   //*********************************************************************
   //*********   ERRFILE IS THE OUTPUT FILE OF REJECTED         ********
   //*********   INFILE RECORDS.                                ********
   //*********************************************************************
   //*********************************************************************
   //*********   AS OF GEOSUPPORT VERSION 10.0,                 ********
   //*********   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G.  ********
   //*********   GRID, PAD, TABFILE ETC) ARE NO LONGER NEEDED   ********
   //*********   AND ARE IGNORED.  GEOSUPPORT IS TAILORED TO    ********
   //*********   USE STANDARD GEOSUPPORT DATA SET NAMES.        ********
   //*********   TO USE NON-STANDARD FILES, PLEASE SEE YOUR     ********
   //*********   SYSTEMS PROGRAMMER.                            ********
   //*********************************************************************
```

```
         //*******************************************************************
 STMT NO. MESSAGE
       3 IEFC001I PROCEDURE GBAT2 WAS EXPANDED USING PRIVATE LIBRARY CSC.TEST.PROCLIB
ICH70001I YOURUID  LAST ACCESS AT 10:27:49 ON FRIDAY, MMMM DD, YYYY
IEF236I ALLOC. FOR EXAMPLE1 GBAT2 S1
IGD103I SMS ALLOCATED TO DDNAME STEPLIB
IGD103I SMS ALLOCATED TO DDNAME
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSTERM
IEF237I JES2 ALLOCATED TO CARDIN
IEF237I JES2 ALLOCATED TO INFILE
IGD101I SMS ALLOCATED TO DDNAME (OUTFILE )
        DSN (SYS06195.T102919.RA000.EXAMPLE1.OUT1A.H01   )
        STORCLAS (PRIMARY) MGMTCLAS (        ) DATACLAS (        )
        VOL SER NOS= SMST01
IEF237I JES2 ALLOCATED TO ERRFILE
IEF237I DMY  ALLOCATED TO ALIASES
GBI SUCCESSFULLY LOADED GBIDRV
GBIDRV (VERSION VV.V) INVOKED
GEO (VERSION VV.V) INVOKED
snd NNN  OPENED SUCCESSFULLY
IGD103I SMS ALLOCATED TO DDNAME SYS00001
PAD NNN  'B030.GEO.COW.BLDGS.CITY' OPENED SUCCESSFULLY
IEF142I EXAMPLE1 GBAT2 S1 - STEP WAS EXECUTED - COND CODE 0000
IGD104I A030.GEO.SUPPORT.PDSE.LOADLIB                RETAINED,  DDNAME=STEPLIB
IGD104I A030.GEO.SUPPORT.LOADLIB                     RETAINED,  DDNAME=
IEF285I   YOURUID.EXAMPLE1.JOB17476.D0000103.?         SYSOUT
IEF285I   YOURUID.EXAMPLE1.JOB17476.D0000104.?         SYSOUT
IEF285I   YOURUID.EXAMPLE1.JOB17476.D0000101.?         SYSIN
IEF285I   YOURUID.EXAMPLE1.JOB17476.D0000102.?         SYSIN
IEF285I   YOURUID.EXAMPLE1.JOB17476.D0000105.?         SYSOUT
IGD104I B030.GEO.COW.BLDGS.CITY                      RETAINED,  DDNAME=SYS00001
IEF373I STEP/GBAT2   /START 2006195.1029
IEF374I STEP/GBAT2   /STOP  2006195.1029 CPU    0MIN 00.06SEC SRB    0MIN 00.00SEC VIRT   928K SYS   308K EXT    8768K SYS   11284K
IEF237I E001 ALLOCATED TO SYS00002
IEF285I   SYS06195.T102920.RA000.EXAMPLE1.R0170302     KEPT
IEF285I   VOL SER NOS= SMST01.
IGD105I SYS06195.T102919.RA000.EXAMPLE1.OUT1A.H01    DELETED,   DDNAME=OUTFILE
IEF375I  JOB/EXAMPLE1/START 2006195.1029
IEF376I  JOB/EXAMPLE1/STOP  2006195.1029 CPU    0MIN 00.06SEC SRB    0MIN 00.00SEC
```

```
*****************************************************************************************************************************
 *****  NOTE:  THIS IS PART OF THE SYSPRINT OUTPUT
 *****************************************************************************************************************************

 USER CONTROL CARDS:

 BORO=9,1  RECTYPE=1A ONSTREET=15,20 1ABLVERSION=S
 TITLE=THIS IS GBAT CONTROL FILE MSW EXAMPLE 1;

 WARNING: CONTROL ENTRIES FOR BOROUGH CODES ARE MISSING - ASSUMED VALUES FOLLOW.
 WARNING: GEOCODE IS MISSING. A DEFAULT VALUE OF YES IS IN EFFECT.
 WARNING: HNI IS MISSING OR UNDEFINED. A DEFAULT VALUE OF NO  IS IN EFFECT.
 WARNING: SNL IS MISSING. A DEFAULT VALUE OF 32 IS IN EFFECT.
 WARNING: MAXREJECTS IS MISSING. A DEFAULT VALUE OF 200 IS IN EFFECT.
 WARNING: REJECTWARNINGS IS MISSING. A DEFAULT VALUE OF NO IS IN EFFECT.
 WARNING: ALIASES IS MISSING. A DEFAULT VALUE OF NO IS IN EFFECT.

 PARAMETERS BEING USED:
 BOROUGH STARTS IN               9               FOR A LENGTH OF                 1
 STREET 1 STARTS IN             15               FOR A LENGTH OF                20
 NORMALIZED STREET LENGTH:                              32
 THE VALUE OF 1ABLVERSION IS:                    S
 RECORD TYPE SPECIFIED: FUNCTION 1A
 THE VALUE OF GEOCODE IS:                        YES
 THE VALUE OF ALIASES IS:                        NO
 THE VALUE OF HNI IS:    NO
 BOROUGH CODE FOR MANHATTAN IS:                  1
 BOROUGH CODE FOR THE BRONX IS:                  2
 BOROUGH CODE FOR BROOKLYN IS:                   3
 BOROUGH CODE FOR QUEENS IS:                     4
 BOROUGH CODE FOR STATEN ISLAND IS:              5
```

```
*****************************************************************************************************************************
 *****  NOTE:  THIS IS PART OF THE SYSPRINT OUTPUT
 *****************************************************************************************************************************

 ***************************************************    STATISTICS  ***********************************************************

  THIS IS GBAT CONTROL FILE MSW EXAMPLE 1                                                                          MM/DD/YY

                                              GEOSUPPORT BATCH ADDRESS TRANSLATOR

                                               MANHATTAN          BRONX        BROOKLYN         QUEENS     STATEN IS.         TOTAL

 INPUT RECORDS                                        10              3               0              1              0            14(*)

 ACCEPTED RECORDS                                      8              2               0              1              0            11

 REJECTED RECORDS:

   28 - A PARTIAL STREET NAME MAY NOT BE USED IN A
        FREE-FORM ADDRESS:                             2              0               0              0              0             2

   42 - ADDRESS NUMBER OUT OF RANGE                    0              1               0              0              0             1

 TOTAL REJECTED RECORDS EXCEPT CODES 17 AND 99:
                                                       2              1               0              0              0             3

   17+99 - BLANK AND INVALID BOROUGH CODES                                                                                        0

 TOTAL REJECTED RECORDS                                                                                                           3

   (*) NOTE  - THIS TOTAL INCLUDES RECORDS WITH INVALID BOROUGH CODES
```

```
*****************************************************************************************************************************
 *****  NOTE:  THIS IS A PRINTOUT OF ERRFILE.  THE FIRST FOUR BYTES CONSIST OF THE TWO-BYTE GEOSUPPORT RETURN CODE (GRC),
 *****         FOLLOWED BY A DASH ('-'), FOLLOWED BY A ONE-BYTE REASON CODE, IF ANY.  IN THIS EXAMPLE, THERE ARE THREE REJECTED
 *****         RECORDS.  TWO HAVE A GRC VALUE OF '28' AND NO REASON CODE VALUE.  THE THIRD REJECT HAS A GRC VALUE OF '42' AND
 *****         NO REASON CODE VALUE.  REFER TO THE GBAT STATISTICS REPORT OR TO APPENDIX 4 FOR THE MESSAGES CORRESPONDING TO
 *****         THE OCCURRING GRC'S.  AFTER THE FIRST FOUR BYTES, THE REST OF THE ERRFILE RECORD CONSISTS OF A COPY OF THE
 *****         REJECTED INFILE RECORD.
******************************************************************************************************************************

 28-        1     60  READE              ** PARTIAL STREET NAMES NOT ALLOWED
 28-        1     12  ELK                ** IN FREE-FORM ADDRESSES
 42-        2     709 E 165 ST
```

SAMPLE GBAT JOB #2

SAMPLE GBAT JOB 2: DESCRIPTION

The control file for Sample GBAT Job 2 is as follows:

```
ALIASES=YES
TITLE=THIS IS GBAT CONTROL FILE EXAMPLE 2;
RECTYPE=2
BORO=5,2
MANHATTAN=MN
BRONX=BX
BROOKLYN=BK
QUEENS=QN
STATEN=SI
ONSTREET=8,25
CROSS1=33,25
GEOCODE=ALL
COMPASS=65,1
COMPACT=YES
REJECTWARNINGS=YES
MAXREJECTS=75
```

In this example, the user has chosen to code each control entry in a separate control record. The user has chosen to align the control entries vertically for aesthetic reasons, although the positioning of each control entry within its control record and the order in which the control entries is codes are immaterial.

The control file in this example contains the following control entries. (See Table A9-1 for Control Entry Descriptions.)

- Since ALIASES=YES has been coded, the user must provide an ALIASES file in the required format (described in Section IX.6), and must provide a DD statement in the JCL referring to that file. GBAT will validate the user's ALIASES file, and will then use the valid street name aliases it contains when processing INFILE. Any invalid aliases will be ignored when processing INFILE, but will be reported in SYSPRINT.

- The TITLE control entry specifies a title for the SYSPRINT output report. Notice that the text of the title is terminated with a semicolon character, as required. (The semicolon does not appear in the actual report.)

- The control entry RECTYPE specifies Function 2.

- The control entry BORO specifies that the input borough code field is in position 5 of the INFILE records and is two bytes long. The input borough code values in this example are not the standard Geosupport borough codes, but are specified as user-defined two-character alphabetic borough codes, 'MN', 'BX', etc., as shown.

- Function 2 requires two input street fields, which in this example are in the form of street names rather than street codes. These fields are specified using the control entries ONSTREET and CROSS1, which state that these fields begin in positions 8 and 33 of the INFILE records, and that each field is 25 bytes long.

- GEOCODE=ALL has been specified, so GBAT will issue a two-work-area call to Function 2 and will form the OUTFILE records by appending the normalized street names and street codes, as well as Work Area 2, to the successfully processed INFILE records.

- The control entry COMPASS specifies an INFILE field for an input compass direction. (Function 2 requires an input compass direction for intersections that are specified in terms of a pair of streets that intersect twice (see Section VII.2). If INFILE contains no such intersections, the control entry COMPASS is not required.)

- COMPACT=YES has been specified, directing GBAT to return all normalized street names in the compact format, which is suitable for display but not for use in sorting.

- REJECTWARNINGS=YES directs GBAT to treat warnings as rejects: they will be written to ERRFILE instead of OUTFILE, they will be counted as errors in the SYSPRINT report, and they will count towards the MAXREJECTS termination limit.

- Since MAXREJECTS=75 has been coded, GBAT will terminate with an MVS Return Code of '20' if the first 75 INFILE records all result in rejects or warnings for any reason other than an invalid borough code.

The user has chosen not to code the following control entries, the default values for which are therefore in effect: (See Table A9-2 for Control Entry default values.)

- Since the control entry VSAM has not been coded, GBAT will assume that the user input data file is a sequential file. Consequently, GBAT will access the input data file via the DDname INFILE, and the corresponding DD statement in the JCL must be coded accordingly.

- Since the SNL control entry has not been coded, GBAT will assume the default value of SNL=32 when normalizing street names.

SAMPLE GBAT JOB 2: JOB-STREAM INPUT

```
//EXAMPLE2 JOB YOUR-JOB-CARD-INFORMATION
//*
//*
//*******************************************************************
//************     THIS JOB IS GBAT MSW EXAMPLE 2     ***************
//*******************************************************************
//*******************************************************************
//*********   THIS STEP INVOKES THE STANDARD CATALOGUED     *********
//*********   PROCEDURE FOR GBAT EXECUTION, CALLED GBAT2    *********
//*******************************************************************
//S1 EXEC GBAT2
//*******************************************************************
//*********   CARDIN IS THE USER-PROVIDED CONTROL FILE       ********
//*******************************************************************
//CARDIN DD *
ALIASES=YES
TITLE=THIS IS GBAT CONTROL FILE MSW EXAMPLE 2;
BORO=5,2
RECTYPE=2
MANHATTAN=MN
BRONX=BX
BROOKLYN=BK
QUEENS=QN
STATEN=SI
ONSTREET=8,25
CROSS1=33,25
GEOCODE=ALL
COMPASS=65,1
COMPACT=YES
REJECTWARNINGS=YES
MAXREJECTS=75
/*
//*******************************************************************
//*********   INFILE IS THE USER-PROVIDED INPUT DATA.          ******
//*********   IN THIS EXAMPLE, IT IS PROVIDED AS INSTREAM DATA.******
//*******************************************************************
//INFILE DD *
    MN READE ST                 BROADWAY
    MN REED ST                  BROADWAY
    MN CANAL ST                 ALLEN ST                        E
    MN CANEL ST                 ALLEN ST                        E
    MN CANAL ST                 ALEN ST                         E
    MN CANEL ST                 ALEN ST                         S
    BK ASSEMBLY RD              GEE AV
    BK ASEMBLY RD               GEE AV
    BK ASSEMBLY RD              GE AV
    BK ASEMBLY RD               GE AV
    MN MAIN ST                  RIVER RD                        S
    MN MAN ST                   RIVER RD                        S
    MN MAIN ST                  RIVE RD                         S
    MN MAN ST                   RIVE RD
    SI HAVEN ESPLN              SILVER LAKE RD
    SI HAVEN ESPLN              SILVER LAKE RD                  N
    QN 116 ST                   CURZON RD                       S
    BX MARINE ST                CITY ISLAND AV                  N
    BX MARINE ST                CITY ISLAND AV
    BX PAULDING AV              SACKET AV                       N
    BK FLATBUSH AV              BEVERLEY RD                     S
    QN QUEENS BL                64 ST                           S
    QN ALDERTON ST              CROMWELL CR                     E
    QN BURDEN CR                84 DR                           N
    BX SHERIF S BYRD PL         JESUP AV
    BX FR MARTIN DOLAN PL       GLEBE AV
/*
```

```
//****************************************************************
//*********   OUTFILE IS THE OUTPUT FILE OF SUCCESSFULLY    ********
//*********   PROCESSED INFILE RECORDS.                     ********
//****************************************************************
//OUTFILE DD DSN=&&OUT,DISP=(NEW,PASS),UNIT=SYSDA,
//        SPACE=(TRK,(80,20),RLSE),
//        DCB=(RECFM=FM,LRECL=364)
//****************************************************************
//*********   ERRFILE IS THE OUTPUT FILE OF REJECTED        ********
//*********   INFILE RECORDS.                               ********
//****************************************************************
//ERRFILE DD SYSOUT=A,DCB=(RECFM=FB,LRECL=84)
//****************************************************************
//*********   ALIASES IS THE OPTIONAL FILE OF USER-PROVIDED ********
//*********   TEMPORARY STREET NAME ALIASES.                ********
//****************************************************************
//ALIASES DD *
1REED ST                        READE ST
1E ST                           ELK ST
1CANEL ST                       CANAL ST
1ALEN ST                        ALLEN ST
3ASEMBLY RD                     ASSEMBLY RD
3GE AV                          GEE AV
1MAN ST                         MAIN ST
1DUANE ST                       DUANE ST
1RIVE RD                        RIVER RD
1RIV RD                         RIVAR RD
1FASHION AVE                    7 AVE
/*
//****************************************************************
//*********   AS OF GEOSUPPORT VERSION 10.0,                ********
//*********   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. ********
//*********   GRID, PAD, TABFILE ETC) ARE NO LONGER NEEDED  ********
//*********   AND ARE IGNORED.  GEOSUPPORT IS TAILORED TO   ********
//*********   USE STANDARD GEOSUPPORT DATA SET NAMES.       ********
//*********   TO USE NON-STANDARD FILES, PLEASE SEE YOUR    ********
//*********   SYSTEMS PROGRAMMER.                           ********
//****************************************************************
//
```

SAMPLE GBAT JOB 2: OUTPUT

```
                    J E S 2  J O B  L O G  --  S Y S T E M  M V S P  --  N O D E  C S C B A T C H

10.31.18 JOB17538 ---- FRIDAY,    DD MMMM YYYY ----
10.31.18 JOB17538  IRR010I  USERID YOURUID  IS ASSIGNED TO THIS JOB.
10.31.18 JOB17538  ICH70001I YOURUID  LAST ACCESS AT 10:30:31 ON FRIDAY, MMMM DD, YYYY
10.31.18 JOB17538  $HASP373 EXAMPLE2 STARTED - INIT 84   - CLASS X - SYS MVSP
10.31.18 JOB17538  IEF403I EXAMPLE2 - STARTED - TIME=10.31.18
10.31.18 JOB17538  +GBI SUCCESSFULLY LOADED GBIDRV
10.31.18 JOB17538  +GBIDRV (VERSION VV.V) INVOKED
10.31.18 JOB17538  +GEO (VERSION VV.V) INVOKED
10.31.18 JOB17538  +snd NNN  OPENED SUCCESSFULLY
10.31.19 JOB17538  +GRID2 NNN  'B030.GEO.COW.GRID2' OPENED SUCCESSFULLY
10.31.19 JOB17538  -                                                --TIMINGS (MINS.)--            ----PAGING COUNTS---
10.31.19 JOB17538  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK   SERV  PG  PAGE  SWAP   VIO SWAPS
10.31.19 JOB17538  -EXAMPLE2 S1       GBAT2       00   1142    180    .00    .00     .0   1264   0     0     0     0     0
10.31.19 JOB17538  IEF404I EXAMPLE2 - ENDED - TIME=10.31.19
10.31.19 JOB17538  -EXAMPLE2 ENDED.  NAME-YOURUID             TOTAL TCB CPU TIME=   .00  TOTAL ELAPSED TIME=    .0
10.31.19 JOB17538  $HASP395 EXAMPLE2 ENDED

------ JES2 JOB STATISTICS ------


  DD MMMM YYYY JOB EXECUTION DATE


          102 CARDS READ


          242 SYSOUT PRINT RECORDS


            0 SYSOUT PUNCH RECORDS


           17 SYSOUT SPOOL KBYTES


         0.01 MINUTES EXECUTION TIME
```

```
1 //EXAMPLE2 JOB YOUR-JOB-CARD-INFORMATION
  //*
  //*
  //*******************************************************************
  //************     THIS JOB IS MSW GBAT EXAMPLE 2      ***************
  //*******************************************************************
  //*******************************************************************
  //*********   THIS STEP INVOKES THE STANDARD CATALOGUED      *********
  //*********   PROCEDURE FOR GBAT EXECUTION, CALLED GBAT2     *********
  //*******************************************************************
3 //S1 EXEC GBAT2
4 XXGBAT2 PROC                                                                00000100
  XX**                        /* IN CSC.TEST.PROCLIB */                       00000200
  XX**                        /* MODIFIED 06/30/06  BY MEB   */               00000315
  XX**                        /* ADDED SUPPORT.PDSE.LOADLIB  */               00000415
  XX**                        /* REMOVED DD CARDS            */               00000515
  XX**                        /* MODIFIED 05/11/06  BY MEB   */               00000615
  XX**                        /* ADDED GRID1R FILE           */               00000715
  XX**                        /* MODIFIED 07/25/05  BY MEB   */               00000815
  XX**                        /* PEDFILE BECOMES DUMMY FILE  */               00000915
  XX**                        /* MODIFIED 03/26/02  BY MEB   */               00001015
  XX**                                                                        00001115
  XX**** WARNING: DO NOT OVERRIDE THE REGION PARAMETER  **************        00001215
  XX**                                                                        00001315
5 XXGBAT2 EXEC PGM=GBATIO2,REGION=9M,PARM='ISASIZE(40K)'                      00001415
  XX*                                                                         00001516
  XX***************************************************************/          00001616
  XX*                                                              */         00001716
  XX*   AS OF GEOSUPPORT VERSION 10.0,                             */         00001816
  XX*   THE STEPLIB (OR JOBLIB) OF THE GEOSUPPORT EXECUTION STEP   */         00001916
  XX*   MUST INCLUDE THE FOLLOWING TWO CONCATENATED DATASETS:      */         00002016
  XX*               A030.GEO.SUPPORT.PDSE.LOADLIB                  */         00002116
  XX*               A030.GEO.SUPPORT.LOADLIB                       */         00002216
  XX*                                                              */         00002316
  XX***************************************************************/          00002416
  XX*                                                                         00002516
  XX*                                                                         00002616
6 XXSTEPLIB DD DSN=A030.GEO.SUPPORT.PDSE.LOADLIB,DISP=SHR                     00002716
7 XX        DD DSN=A030.GEO.SUPPORT.LOADLIB,DISP=SHR                          00002816
  XX*                                                                         00002916
  XX*                                                                         00003016
  XX***************************************************************/          00003116
  XX*                                                              */         00003216
  XX*   AS OF GEOSUPPORT VERSION 10.0,                             */         00003316
  XX*   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G. GRID, PAD    */         00003416
  XX*   ETC) ARE NO LONGER NEEDED AND ARE IGNORED.  GEOSUPPORT     */         00003516
  XX*   IS TAILORED TO USE STANDARD GEOSUPPORT DATA SET NAMES.     */         00003616
  XX*   TO USE NON-STANDARD FILES, SEE YOUR SYSTEMS PROGRAMMER.    */         00003716
  XX*                                                              */         00003816
  XX***************************************************************/          00003916
  XX*                                                              */         00004016
  XX*                                                              */         00004116
8 XXSYSPRINT DD SYSOUT=A,DCB=(LRECL=133,RECFM=FBA,BLKSIZE=1330)               00004216
  XX** SYSPRINT FILE CONTAINS RUN STATISTICS AND MESSAGES                     00004316
9 XXSYSTERM  DD SYSOUT=A,DCB=(LRECL=133,RECFM=FBA,BLKSIZE=1330)               00004416
```

```
         XX** SYSTERM FILE CONTAINS SYSTEM WARNINGS AND ERRORS                    00004516
      10 //CARDIN DD *
         X/CARDIN DD DDNAME=CARDIN                                                00004616
         XX** CARDIN IS THE FILE OF GBAT CONTROL RECORDS                          00004716
      11 //INFILE DD *
         X/INFILE DD DDNAME=INFILE                                                00004816
         XX** INFILE CONTAINS THE USERS DATA INPUT RECORDS                        00004916
      12 //OUTFILE DD DSN=&&OUT,DISP=(NEW,PASS),UNIT=SYSDA,
         //         SPACE=(TRK,(80,20),RLSE),
         //         DCB=(RECFM=FM,LRECL=364)
         X/OUTFILE DD DDNAME=OUTFILE                                              00005016
         XX** OUTFILE CONTAINS THE VALID OUTPUT RECORDS                           00005116
      13 //ERRFILE DD SYSOUT=A,DCB=(RECFM=FB,LRECL=84)
         X/ERRFILE DD DDNAME=ERRFILE                                              00005216
         XX** ERRFILE CONTAINS THE REJECTS                                        00005316
      14 //ALIASES DD *
         X/ALIASES  DD DUMMY                                                      00005416
         XX** ALIASES IS THE OPTIONAL INPUT FILE OF USER-DEFINED ST NAME ALIASES 00006015
         //*******************************************************************
         //*********   CARDIN IS THE USER-PROVIDED CONTROL FILE       ********
         //*******************************************************************
         //*******************************************************************
         //*********   INFILE IS THE USER-PROVIDED INPUT DATA.          ******
         //*********   IN THIS EXAMPLE, IT IS PROVIDED AS INSTREAM DATA.******
         //*******************************************************************
         //*******************************************************************
         //*********   OUTFILE IS THE OUTPUT FILE OF SUCCESSFULLY     ********
         //*********   PROCESSED INFILE RECORDS.                      ********
         //*******************************************************************
         //*******************************************************************
         //*********   ERRFILE IS THE OUTPUT FILE OF REJECTED         ********
         //*********   INFILE RECORDS.                                ********
         //*******************************************************************
         //*******************************************************************
         //*********   ALIASES IS THE OPTIONAL FILE OF USER-PROVIDED  ********
         //*********   TEMPORARY STREET NAME ALIASES.                 ********
         //*******************************************************************
         //*******************************************************************
         //*********   AS OF GEOSUPPORT VERSION 10.0,                 ********
         //*********   DD STATEMENTS FOR GEOSUPPORT DATA FILES (E.G.  ********
         //*********   GRID, PAD, TABFILE ETC) ARE NO LONGER NEEDED   ********
         //*********   AND ARE IGNORED.  GEOSUPPORT IS TAILORED TO    ********
         //*********   USE STANDARD GEOSUPPORT DATA SET NAMES.        ********
         //*********   TO USE NON-STANDARD FILES, PLEASE SEE YOUR     ********
         //*********   SYSTEMS PROGRAMMER.                            ********
         //*******************************************************************
 STMT NO. MESSAGE
        3 IEFC001I PROCEDURE GBAT2 WAS EXPANDED USING PRIVATE LIBRARY CSC.TEST.PROCLIB
ICH70001I YOURUID  LAST ACCESS AT 10:30:31 ON FRIDAY, MMMM DD, YYYY
IEF236I ALLOC. FOR EXAMPLE2 GBAT2 S1
IGD103I SMS ALLOCATED TO DDNAME STEPLIB
IGD103I SMS ALLOCATED TO DDNAME
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSTERM
IEF237I JES2 ALLOCATED TO CARDIN
```

```
IEF237I JES2 ALLOCATED TO INFILE
IGD101I SMS ALLOCATED TO DDNAME (OUTFILE )
        DSN (SYS06195.T103118.RA000.EXAMPLE2.OUT.H01     )
        STORCLAS (PRIMARY) MGMTCLAS (        ) DATACLAS (        )
        VOL SER NOS= SMST07
IEF237I JES2 ALLOCATED TO ERRFILE
IEF237I JES2 ALLOCATED TO ALIASES
GBI SUCCESSFULLY LOADED GBIDRV
GBIDRV (VERSION VV.V) INVOKED
GEO (VERSION VV.V) INVOKED
snd NNN  OPENED SUCCESSFULLY
IGD103I SMS ALLOCATED TO DDNAME SYS00001
GRID2 NNN  'B030.GEO.COW.GRID2' OPENED SUCCESSFULLY
IEF142I EXAMPLE2 GBAT2 S1 - STEP WAS EXECUTED - COND CODE 0000
IGD104I A030.GEO.SUPPORT.PDSE.LOADLIB                RETAINED,  DDNAME=STEPLIB
IGD104I A030.GEO.SUPPORT.LOADLIB                     RETAINED,  DDNAME=
IEF285I   YOURUID.EXAMPLE2.JOB17538.D0000104.?         SYSOUT
IEF285I   YOURUID.EXAMPLE2.JOB17538.D0000105.?         SYSOUT
IEF285I   YOURUID.EXAMPLE2.JOB17538.D0000101.?         SYSIN
IEF285I   YOURUID.EXAMPLE2.JOB17538.D0000102.?         SYSIN
IEF285I   YOURUID.EXAMPLE2.JOB17538.D0000106.?         SYSOUT
IEF285I   YOURUID.EXAMPLE2.JOB17538.D0000103.?         SYSIN
IGD104I B030.GEO.COW.GRID2                           RETAINED,  DDNAME=SYS00001
IEF373I STEP/GBAT2   /START 2006195.1031
IEF374I STEP/GBAT2   /STOP  2006195.1031 CPU    0MIN 00.06SEC SRB    0MIN 00.00SEC VIRT  1020K SYS   300K EXT    8748K SYS   11384K
IEF237I E901 ALLOCATED TO SYS00002
IEF285I   SYS06195.T103119.RA000.EXAMPLE2.R0170325     KEPT
IEF285I   VOL SER NOS= SMST07.
IGD105I SYS06195.T103118.RA000.EXAMPLE2.OUT.H01      DELETED,   DDNAME=OUTFILE
IEF375I  JOB/EXAMPLE2/START 2006195.1031
IEF376I  JOB/EXAMPLE2/STOP  2006195.1031 CPU    0MIN 00.06SEC SRB    0MIN 00.00SEC
```

```
*****************************************************************************************************************************
***** NOTE:  THIS IS PART OF THE SYSPRINT OUTPUT
*****************************************************************************************************************************

USER CONTROL CARDS:

ALIASES=YES
TITLE=THIS IS GBAT CONTROL FILE MSW EXAMPLE 2;
BORO=5,2
RECTYPE=2
MANHATTAN=MN
BRONX=BX
BROOKLYN=BK
QUEENS=QN
STATEN=SI
ONSTREET=8,25
CROSS1=33,25
GEOCODE=ALL
COMPASS=65,1
COMPACT=YES
REJECTWARNINGS=YES
MAXREJECTS=75

WARNING: SNL IS MISSING. A DEFAULT VALUE OF 32 IS IN EFFECT.

PARAMETERS BEING USED:
BOROUGH STARTS IN              5               FOR A LENGTH OF                 2
STREET 1 STARTS IN             8               FOR A LENGTH OF                25
STREET 2 STARTS IN            33               FOR A LENGTH OF                25
NORMALIZED STREET LENGTH:                              32
COMPASS STARTS IN             65               FOR A LENGTH OF                 1
COMPACT OPTION   WAS SPECIFIED
RECORD TYPE SPECIFIED: FUNCTION 2
THE VALUE OF GEOCODE IS:                       ALL
WARNINGS ARE TREATED AS REJECTS
MAXIMUM NUMBER OF REJECTS ALLOWED IS                        75
THE VALUE OF ALIASES IS:                       YES
BOROUGH CODE FOR MANHATTAN IS:                 MN
BOROUGH CODE FOR THE BRONX IS:                 BX
BOROUGH CODE FOR BROOKLYN IS:                  BK
BOROUGH CODE FOR QUEENS IS:                    QN
BOROUGH CODE FOR STATEN ISLAND IS:             SI
ERROR: ALIASES INPUT RECORD NUMBER 0008 HAS BEEN REJECTED.
ALIAS STREET NAME AND STREET NAME RECOGNIZED BY GEOSUPPORT ARE INDENTICAL - DUANE ST                        .

ERROR: ALIASES INPUT RECORD NUMBER 0010 HAS BEEN REJECTED:
RETURN CODE = EE FOR STNAME RIVAR RD                        .
ERROR: ALIASES INPUT RECORD NUMBER 0011 HAS BEEN REJECTED.
FASHION AVE                     AND 7 AVE                         HAVE DIFFERENT 7 DIGIT STCODES (11061002 AND 11061004).

NOTE:  ALIAS TABLE HAS ERRORS AND ALIASES=YES HAS BEEN SPECIFIED-PROCESSING CONTINUED.
```

```
**************************************************************************************************************************************
 ***** NOTE:  THIS IS PART OF THE SYSPRINT OUTPUT
 **************************************************************************************************************************************

 *****************************************************    STATISTICS   *************************************************************

  THIS IS GBAT CONTROL FILE MSW EXAMPLE 2                                                                                   MM/DD/YY

                                                     GEOSUPPORT BATCH ADDRESS TRANSLATOR

                                                      MANHATTAN       BRONX    BROOKLYN      QUEENS  STATEN IS.     BOROUGH       TOTAL
                                                                                                                    BOUNDARY

 INPUT RECORDS                                               10           5           5           4           2           0       26(*)

 ACCEPTED RECORDS                                             8           4           5           2           0           0       19

 REJECTED RECORDS:

   01 - WARNING MESSAGES                                      0           0           0           1           0           0        1

   02 - THESE STREETS INTERSECT TWICE-COMPASS
        DIRECTION REQUIRED:                                   1           1           0           0           0           0        2

   03 - THESE STREETS INTERSECT MORE THAN
        TWICE-CANNOT BE PROCESSED:                            0           0           0           1           2
0           3

   40 - COMPASS DIRECTION VALUE IS INVALID FOR THIS
        INPUT LOCATION:                                       1           0           0           0           0           0        1

 TOTAL REJECTED RECORDS EXCEPT CODES 17 AND 99:
                                                              2           1           0           2           2           0        7

   17+99 - BLANK AND INVALID BOROUGH CODES                                                                                         0

 TOTAL REJECTED RECORDS                                                                                                            7

   (*) NOTE  - THIS TOTAL INCLUDES RECORDS WITH INVALID BOROUGH CODES
```

```
*******************************************************************************************************************************
***** NOTE:  THIS IS A PRINTOUT OF ERRFILE.  THE FIRST FOUR BYTES CONSIST OF THE TWO-BYTE GEOSUPPORT RETURN CODE (GRC),
*****        FOLLOWED BY A DASH ('-'), FOLLOWED BY A ONE-BYTE REASON CODE, IF ANY.  IN THIS EXAMPLE, THERE ARE SEVEN REJECTED
*****        RECORDS.  TWO HAVE A GRC VALUE OF '02' AND NO REASON CODE VALUE.  TWO HAVE A GRC VALUE OF '03' AND A REASON CODE
*****        VALUE OF '3'.  ONE HAS A GRC VALUE OF '01' (WARNING) AND A REASON CODE VALUE OF 'H', ETC.  REFER TO THE GBAT
*****        STATISTICS REPORT OR TO APPENDIX 4 FOR THE MESSAGES CORRESPONDING TO THE OCCURRING GRC'S AND REASON CODES.  AFTER
*****        THE FIRST FOUR BYTES, THE REST OF THE ERRFILE RECORD CONSISTS OF A COPY OF THE REJECTED INFILE RECORD.
*******************************************************************************************************************************

40-     MN CANEL ST                 ALEN ST                         S
02-     MN MAN ST                   RIVE RD
03-3    SI HAVEN ESPLN              SILVER LAKE RD
03-3    SI HAVEN ESPLN              SILVER LAKE RD                  N
01-H    QN 116 ST                   CURZON RD                       S
02-     BX MARINE ST                CITY ISLAND AV
03-4    QN QUEENS BL                64 ST                           S
```

APPENDIX 11: GUIDELINES FOR APPLICATION DESIGN

This appendix contains guidelines for application designers, listed in no particular order. These guidelines are intended only to be a limited selection of helpful suggestions, not a comprehensive set of instructions for application design. Terms highlighted in **bold typeface** have entries in the Glossary.

2) DESIGN PROCEDURES TO REVIEW AND (WHEN APPROPRIATE) TO REPORT **REJECTS** TO **GSS**: As an integral part of the application, set up procedures to examine geographic data that have been rejected by Geosupport, and to report appropriate rejects to GSS. Only those rejected data that, after examination, do not appear attributable to user errors should be reported to GSS. In addition, users should also report cases in which the input information was not rejected, but the output data that Geosupport has returned to the application appear to be erroneous for the given location (such as an incorrect zipcode or incorrect cross streets). User feedback is essential to GSS's efforts to keep Geosupport accurate and up to date.

3) USE THE GEOSUPPORT **COPY LIBRARIES**: If the application is being written in a programming language supported by Geosupport's **COPY facility** (currently, COBOL, PL/1, IBM Mainframe Assembler Language, C or NATURAL), do not code layouts of the Geosupport **work areas** directly into the application program source code. Instead, write the program to access the Geosupport **COPY facility.** This will cause the program to automatically obtain the most current standard source code work area layouts at compile time. This approach eliminates tedious and error-prone line-by-line coding of the Geosupport work area layouts by the application programmer, insures the use of the most current layouts, and facilitates trouble-shooting by insuring the use of standard data names for Geosupport data items.

4) DESIGN FOR **GEOGRAPHIC RETRIEVAL CONSISTENCY**: If an application is required to retrieve data from the application's own files by geographic location, it should be designed so that it performs such retrieval consistently, that is, independently of variations in specifying geographic locations. This is accomplished by obtaining certain items from Geosupport, storing them in the application file, and using them as part of the retrieval key. For example, for retrievals by address, use B5SCs instead of street names in the retrieval key. For building-level retrievals, store **BIN**s in the application file and use them rather than addresses or tax lot identifiers as the retrieval key.

5) DESIGN BATCH PROCEDURES TO RE-SYNCHRONIZE APPLICATION FILES WITH NEW GEOSUPPORT RELEASES: Geographic information changes over time. For example, changes are possible in the election districts, tax lot identifiers or police precincts associated with addresses, in the **street codes** assigned to street names, in the streets incident upon intersections, etc. During application design, consider which data items obtained from Geosupport and stored in application files should be updated to

reflect changes in new Geosupport **releases**, and design procedures to perform such updating. In particular, if street codes are to be stored in an application file, store them in the form of B10SCs (but use only the B5SC portions for geographic retrieval), and develop a fully automated batch **resynchronization** procedure utilizing the Street Code Change File (see Section IV.4).

6) USE THE APPROPRIATE STREET NAME FOR THE TASK: To sort a file by geographic location, always use **street names normalized** in **sort format** (see Section III.3). For display purposes, obtain **preferred street names** (see Section IV.6)**,** and display them **normalized** in **compact format** (see Section III.3).

7) WHENEVER POSSIBLE, ALLOCATE 32 BYTES FOR STREET NAME FIELDS; DO NOT SPECIFY AN **SNL** VALUE UNLESS THE APPLICATION SPECIFICALLY REQUIRES SHORTER STREET NAME FIELDS (for example, to fit within a limited amount of space in a report, screen or transparent envelope window). The default SNL value, 32, insures that all valid input street names can be successfully normalized.

8) TO VALIDATE ADDRESSES, USE FUNCTION 1A RATHER THAN FUNCTION 1 OR 1E. Function 1A does a far better job of validating whether a building having a given address actually exists.

9) TO IMPROVE EXECUTION EFFICIENCY, use Function 1 instead of Function1E unless the application requires the political district geography that only Function 1E provides. (Function 1 performs fewer I/O operations.)

10) NEVER DESIGN NEW APPLICATIONS TO USE **VESTIGIAL FEATURES** OF GEOSUPPORT (see Section I.5).

11) ESCHEW **FREE-FORM ADDRESS** PROCESSING (see Section V.3) UNLESS IT IS UNAVOIDABLE. Whenever possible, pass the house number and the street name of an address to Geosupport in the separate **WA1** input fields for those items. Design application files so that the house number and street name of an address are stored in separate fields.

12) REVIEW THE SET OF **WARNING** AND **REJECT** CONDITIONS THAT CAN BE ISSUED BY EACH FUNCTION THE APPLICATION WILL BE CALLING. Determine whether any of these conditions warrant custom handling in your application. Appendices 1 and 4 of this document are useful in this regard.

13) DESIGN INTERACTIVE APPLICATIONS TO USE THE SIMILAR NAMES FEATURE. (See Section III.5.) Whenever Geosupport rejects an input street name and returns similar names, display the list of similar names on the screen and allow (but do not require) the operator to select one of them using the cursor. If the operator selects a

similar name, re-submit the Geosupport call automatically using the similar name in place of the rejected input name.

APPENDIX 12: CHARACTER-ONLY WORK AREAS (COW)

Introduction

This appendix is based on Geosupport System Technical Bulletin 02-01 (dated 15 November 2002) and Geosupport Technical Bulletin 02-01 Addendum (dated 22 November 2002). It contains information needed to create Geosupport applications using the Character-Only Work Areas (COWs). Included are the following topics:

- Comparison of COWs and Mainframe-Specific Work Areas (MSWs)
- Considerations when using COWs
- Work Area Lengths
- Specifying the Work Area Set
- GBAT Considerations
- COW COPY Files

Notes:

1. There are two versions of the sample programs in Appendix 5 of this *User Programming Guide*. One version is for MSWs and the other is for COWs. When coding, bear in mind the differences between COWs and MSWs.

2. The Work Area Layouts in Appendix 2 of this *User Programming Guide* are the layouts of the MSWs. For the COW layouts, see Appendix 13.

3. The COPY files that are printed in Appendix 5 of this *User Programming Guide* are for the Mainframe-Specific Work Areas (MSWs). For the COW COPY files, see Appendix 14.

Overview

Standard work areas with pre-defined layouts are used to pass data between the Geosupport System and user-developed application programs. The same work areas are also used by GBAT, the Geosupport batch utility program, to pass data to and from Geosupport.

The Geosupport work areas that have long been in use are called the Mainframe-Specific Work Areas (MSWs). Most of the MSWs contain one or more packed decimal fields, a data encoding schema unique to IBM mainframes. This appendix discusses an alternative set of Geosupport work areas called the Character-Only Work Areas (COWs) which, as the name implies, contain character fields only. The introduction of the COW is an essential part of a long-term effort to port the Geosupport System to other platforms.

Each specific Geosupport work area (for example, Work Area 2 for Function 3S) has both a COW version and an MSW version. User-written application programs running on mainframes now have the option to use either set of work areas when making calls to Geosupport. GBAT users can also specify the use of either set of work areas.

From now on, all new applications should be designed to use the COWs. We also recommend that **all existing applications be converted to use the COWs**. Although the MSWs will continue to be supported, as of some future date (not yet determined), only the COWs will be enhanced with new data items and functionality. Eventually, the MSWs may be de-supported.

Comparison of COWs and MSWs

Each non-character field in an MSW has a character field counterpart in the corresponding COW. Except for an item called the HND (discussed below in the sub-section on house number fields), each character field in an MSW appears in identical form in the corresponding COW. However, corresponding fields do not necessarily occupy the same byte positions or occur in the same order in the corresponding MSW and COW. In designing the COWs, the opportunity has been taken to reorganize the layouts to situate related fields near each other and to increase the amount of filler space available for adding new data items in the future.

The data items for which the MSWs contain non-character fields are house numbers, street codes, segment lengths, and count fields. Each of these is discussed in detail below.

House Number Fields. The Geosupport System uses three different formats for standardized or 'normalized' house numbers: the House Number in Display format (HND), the House Number in Internal format (HNI), and the new House Number in Sort format (HNS).

- The **HND** is a character item that is present in both the COWs and the MSWs, but it has a different length in each: 16 bytes in the COWs and 12 bytes in the MSWs. The length of the HND was increased in the COWs to insure that house numbers having suffixes fit within the HND field without the suffix having to be abbreviated. (House number suffixes are certain character strings that occur at the ends of some New York City house numbers, such as 1/2,1/4, REAR, GARAGE.) For compatibility with MSW, by default Geosupport uses only the first 12 characters of the 16-byte COW HND. The remaining 4 characters are blank. To use all 16 characters, the user can specify an HNL (House Number Length) of 16.

- The **<u>HNI</u>** is a six-byte data item with a hybrid format: the first five bytes are in packed decimal format, and the sixth byte contains a binary value. HNIs occur only in the MSW.

- The **<u>HNS</u>** is a new 11-byte item that is the character equivalent of the HNI in the COWs.

The HND is the appropriate format for displaying house numbers on application screens, reports and computer-generated maps, and is specifically designed for that purpose. In particular, the HND is left-justified and space-filled. However, the HND renders unsatisfactory results when used as a field to sort addresses. For example, it would, inappropriately, cause 102 MAIN STREET to sort ahead of 98 MAIN STREET.

In contrast to the HND, both the HNI and the HNS are suitable to use as fields to sort addresses. For example, both would, appropriately, cause 98 MAIN STREET to sort ahead of 102 MAIN STREET. However, neither the HNI nor the HNS is suitable for display:

- The HNI is not a character item and so cannot be displayed as intelligible data unless first converted to character format. Any of the Geosupport display functions (Functions D, DG and DN) can be used with the MSWs to convert an HNI to an HND.

- The HNS, although it is a character item, is unsuitable for display. In particular, the HNS is in an internal format with a unique layout and flags. If the house number has a suffix, the HNS does not contain the suffix itself, but instead, contains a code for the suffix meaningful only to the Geosupport software. Any of the Geosupport display functions (Function D, DG and DN) can be used with the COWs to convert an HNS to an HND.

To reiterate, **the HND should be used for display, and the HNI (in MSWs) or the HNS (in COWs) should be used for sorting.**

<u>Street Code Fields</u>. A notable feature of the Geosupport System is its set of numeric street codes assigned to the names of New York City streets and selected non-street geographic features. The street code feature provides specialized capabilities that are essential for certain types of applications.

Street codes appear in several forms in the Geosupport work areas. In many of the MSWs, there are four-byte fields for a packed decimal data item called the Packed Borough and 5-Digit Street Code (PB5SC). The COW counterparts of PB5SC fields can take one of the following three forms:

- A six-byte field for an item called the Borough and 5-Digit Street Code (B5SC). This is simply the unpacked version of the PB5SC

- The first six bytes (constituting the B5SC) of an eight-byte field for an item called the Borough and 7-Digit Street Code (B7SC)

- The first six bytes (constituting the B5SC) of an 11-byte field for an item called the Borough and 10-Digit Street Code (B10SC)

Segment Length Fields. Both Work Area 2 for Function 3 and Work Area 2 for Function 3C contain fields for the segment length expressed in feet. In the MSW format, these are 3-byte packed decimal fields. In the COW format, they are 5-byte character fields.

Count Fields. Some count fields, e.g. Number of Street Names in List, are packed decimal fields in the MSW format, and character fields in the COW format.

Consideration When Using the COWs

The Long Work Area 2 option that is available when using the MSWs is occasionally not needed or not supported when using the COWs, as follows:

- When using the COWs, Functions 1, 1E and 3 do not have the long Work Area 2 option. This option is unnecessary in these cases, since the COW versions of the regular Work Area 2s for these functions already accommodate all the requisite fields. However, Functions 1A and BL continue to have the long Work Area 2 option when COWs are used. (See Section II.5 for a general discussion of the long Work Area 2 option.)

Work Area Lengths (COWs and MSWs)

The following table lists the lengths of the members of both sets of work areas. Note that the lengths of corresponding members from the two sets differ in most cases.

Table A12-1: Lengths of Work Areas (COWs and MSWs)

Work Area	Length of COW	Length of MSW
Work Area 1 (used with all functions)	1200	884
Regular WA2 for Functions 1, 1E	300	200
Long WA2 for Functions 1, 1E	N/A	300
Regular WA2 for Functions 1A, BL, BN	1363	939
Long WA2 for Functions 1A, BL	17750	17683
WA2 for Function 2	200	200
Regular WA2 for Function 3	450	200
Long WA2 for Function 3	N/A	300
WA2 for Function 3C	300	200
WA2 for Function 3S	19274	4224

Specifying a Work Area Set (COW or MSW)

To indicate which set of work areas is being used in a call to Geosupport, an application program uses a new field called the Work Area Format Indicator. This field is one byte long and is located at position 213 of both the COW Work Area 1 and the MSW Work Area 1.

- The value 'C' in the Work Area Format Indicator indicates to Geosupport that COWs are being used for the given call.
- A blank in the Work Area Format Indicator indicates that MSWs are being used.
- If the Work Area Format Indicator is invalid, the call is rejected with a Geosupport Return Code of 27 and an appropriate message.

Since every call to Geosupport is an independent event, application programs must insure that the Work Area Format Indicator is appropriately set for each call; Geosupport doesn't 'remember' previous calls.

Note that the MSWs are the default work areas, that is, the work areas that Geosupport expects when the Work Area Format Indicator is blank. Therefore, **existing applications that use the MSWs will continue to execute properly without modification, provided Work Area 1 is being passed to Geosupport with position 213 containing a blank**. (As a matter of course, every application program should be designed so that, each time a call to Geosupport is to be made, the program clears Work Area 1 entirely to blanks prior to moving the input data for that call into the requisite Work Area 1 fields. This insures that Work Area 1 will not be 'polluted' by stray input data lingering from a previous call.)

GBAT Considerations for COWs

When executing GBAT, the set of work areas that are used affects the length and format of the records written into OUTFILE (the output file of successfully processed data records).

To specify the set of work areas GBAT is to use, the user codes a control entry in CARDIN (the input control file) containing the keyword WORKAREA, as follows:

- WORKAREA=COW specifies the COWs.
- WORKAREA=MSW specifies the MSWs.
- (Default:) If no WORKAREA control entry is coded, GBAT uses the MSWs.

Since the MSWs are GBAT's default set of work areas, existing GBAT jobs will continue to execute properly without modification.

When COWs are used, GBAT options that involve processing packed decimal input data are, of course, invalid. Specifically, the following control entries or control entry variable values are **invalid** when COWs are being used:

- HNI=YES is invalid.
- In the control entries STRTCODE, CRSCOD1 and CRSCOD2, the values 3 and 4 are invalid for the length variable.
- 1ABLVERSION=L (or 1ABLVERSION=LEGACY) is invalid, and is, in fact, no longer supported.

If the 1ABLVERSION control entry is not coded at all, the default value in effect depends on the set of work areas being used:

- If COWs are being used,
the default is 1ABLVERSION=S (or 1ABLVERSION=STANDARD).
- If MSWs are being used,
the default is 1ABLVERSION=L (or 1ABLVERSION=LEGACY), which results in an error, since Legacy is no longer supported.

GBAT forms each OUTFILE record by appending Geosupport information to a copy of the data input record. The information that is appended is determined by three factors: the set of work areas being used, the function being executed and the value of the GEOCODE control entry that is in effect. The GEOCODE value affects the appended information as follows:

- When GEOCODE=NO, the appended items consist only of output items from Work Area 1 (and, in the case of Functions 1, 1A and 1E when the MSWs are being used, an item created by GBAT called the HNHPD). See Table A12-2 below for lists of the appended COW items and their lengths by function and set of work areas. For MSW format, see Tables 9-4 and 9-5.

- When GEOCODE=YES, the appended information consists only of Work Area 2 in its entirety. Table A12-1 above lists the lengths of these work areas by function and work area set. Appendix 2 contains the MSW Work Area Layouts and Appendix 13 contains the COW Work Area Layouts.

- When GEOCODE=ALL, the appended information consists of the data appended for GEOCODE=NO followed by the data appended for GEOCODE=YES. Table A12-3 lists the lengths of the appended information by function and set of work areas. These lengths typically are the sums of the corresponding lengths listed in Tables A12-1 and A12-2.

YES and ALL are invalid GEOCODE options for Functions 1N, D, DG and DN, since these functions do not have a Work Area 2.

The information appended for GEOCODE=NO is as follows.

- <u>For the functions that allow an input house number other than the display functions (viz, Functions 1, 1A and 1E)</u>: the appended information includes normalized house number items as follows:

    - If COWs are being used, the HND and the HNS are appended.

    - If MSWs are being used, the HND and an 8-byte item called the HNHPD are appended. The HNHPD is a normalized house number created by GBAT in a special format for use only by the Department of Housing Preservation and Development.

- <u>For the display functions (Functions D, DG and DN), which allow input house numbers in the form of HNIs when using MSWs</u>: the appended information includes the normalized house numbers only in the HND format for a length of 12.

- <u>For the display functions (Functions D, DG and DN), which allow input house numbers in the form of HNSs when using COWs</u>: the appended information includes the normalized house numbers only in the HND format for a length of between 12 and 16.

- For the functions that involve input street names (Functions 1, 1A, 1E, 1N, 2, 3, 3C, 3S, D, DG and DN): the appended information includes normalized street name(s) and street codes, as follows:

    - Regardless of which set of work areas is being used, normalized street names are provided in 32-byte fields, left-justified and blank-filled.

    - If COWs are being used, street codes are provided as B10SCs, an 11-byte item.

    - If MSWs are being used, street codes are provided as 10SCs (ten-digit street codes without a borough code), a 10-byte item.

- For Function BL: the appended information when 1ABLVERSION=STANDARD is specified consists of the standard 10-byte BBL, which is composed of the one-byte borough code followed by the five-byte tax block followed by the four-byte tax lot. Note: STANDARD is the only valid value for 1ABLVERSION.

- For Function BN: the appended information consists of ten bytes containing the seven-byte Building Identification Number (BIN) followed by a three-byte filler.

Table A12-2 below lists the appended items when GEOCODE=NO, itemized by function and set of work areas. When an item of appended data has no value for a particular record (such as a house number item, when the input location is a non-addressable place name), the given field is still present in the appended data but it contains all blanks. The only exception to this involves the display functions (D, DG and DN) when using MSWs, where there is no field for the second house number unless it is provided as an input datum by the user.

Table A12-2: GBAT-Appended Items when GEOCODE=NO (COWs and MSWs)

Functions	**COWs**		**MSWs**	
[Options]	**Appended Items**	**Length**	**Appended Items**	**Length**
1, 1A, 1E	HND HNS Normalized Street Name B10SC	16 11 32 11	HND HNHPD Normalized Street Name 10SC	12 8 32 10
	Total Length:	70	Total Length:	62

Table A12-2: GBAT-Appended Items when GEOCODE=NO (COWs and MSWs) (cont.)

Functions [Options]	COWs Appended Items	COWs Length	MSWs Appended Items	MSWs Length
1N	Normalized Street Name B10SC	32 11	Normalized Street Name 10SC	32 10
	Total Length:	43	Total Length:	42
2	Normalized Street Name-1 B10SC-1 Normalized Street Name-2 B10SC-2	32 11 32 11	Normalized Street Name-1 10SC-1 Normalized Street Name-2 10SC-2	32 10 32 10
	Total Length:	86	Total Length:	84
3, 3C, 3S	Normalized Street Name-1 B10SC-1 Normalized Street Name-2 B10SC-2 Normalized Street Name-3 B10SC-3	32 11 32 11 32 11	Normalized Street Name-1 10SC-1 Normalized Street Name-2 10SC-2 Normalized Street Name-3 10SC-3	32 10 32 10 32 10
	Total Length:	129	Total Length:	126
BL (Standard)	BBL : Borough Code Tax Block Tax Lot	 1 5 4	BBL : Borough Code Tax Block Tax Lot	 1 5 4
	Total Length:	10	Total Length:	10
BL (Legacy)	Invalid		Invalid	
BN	BIN Filler	7 3	BIN Filler	7 3
	Total Length:	10	Total Length:	10
D, DG, DN [HOUSENUM2 control entry not coded]	HND-1 Normalized Street Name-1 Normalized Street Name-2 Normalized Street Name-3 HND-2	16 32 32 32 16	HND Normalized Street Name-1 Normalized Street Name-2 Normalized Street Name-3	12 32 32 32
	Total Length:	128	Total Length:	108
D, DG, DN [HOUSENUM2 control entry coded]	HND-1 Normalized Street Name-1 Normalized Street Name-2 Normalized Street Name-3 HND-2	16 32 32 32 16	HND-1 Normalized Street Name-1 Normalized Street Name-2 Normalized Street Name-3 HND-2	12 32 32 32 12
	Total Length:	128	Total Length:	120

Table A12-3 lists the length of the data appended by GBAT, itemized by function and option within function, GEOCODE value and set of work areas being used.

Table A12-3: Length of GBAT-Appended Data (COWs and MSWs)

Functions	Options	GEOCODE=NO		GEOCODE=YES		GEOCODE=ALL	
		COWs	MSWs	COWs	MSWs	COWs	MSWs
1, 1E	LONGWA2=NO	70	62	300	200	370	262
	LONGWA2=YES	Invalid	Invalid	Invalid	300	Invalid	362
1A	LONGWA2=NO	70	62	1363	939	1433	1001
	LONGWA2=YES	Invalid	Invalid	17750	17683	17820	17745
1N		43	42	Invalid	Invalid	Invalid	Invalid
2		86	84	200	200	286	284
3	LONGWA2=NO	129	126	450	200	579	326
	LONGWA2=YES	Invalid	Invalid	Invalid	300	Invalid	426
3C		129	126	300	200	429	326
3S		129	126	19274	4224	19403	4350
BL	LONGWA2=NO, 1ABLVERSION=STANDARD	10	10	1363	939	1373	949
	LONGWA2=YES, 1ABLVERSION=STANDARD	Invalid	Invalid	17750	17683	17760	17693
BN		10	10	1363	939	1373	949
D, DG, DN	HOUSENUM2 not coded	128	108	Invalid	Invalid	Invalid	Invalid
D, DG, DN	HOUSENUM2 coded	128	120	Invalid	Invalid	Invalid	Invalid

COPY Files for COWs

For COBOL, PL/1, BAL and C programmers, copy files have been created to enable you to easily use the new work areas (COWs) in your programming work. Local Data Areas will be available for Natural programmers. If you use the MVSP LPAR at DoITT, these items will be found in library ‘A030.GEO.COPYLIB’. If you use any other LPAR at DoITT or if you work at a different data center, contact your system programming staff to learn the name of the library in which these members are stored. The following table shows you the copy file name by language and Geosupport Function.

Table A12-4: List of COPY Files for COWs

Language	Work Area	Copy File
COBOL	Work Area 1 (used with all Functions) Work Area 2 for Functions 1 and 1E Regular Work Area 2 for Functions 1A, BL, BN Long Work Area 2 for Functions 1A and BL Work Area 2 for Function 2 Work Area 2 for Function 3 Work Area 2 for Function 3C Work Area 2 for Function 3S	P1COB P2COB P2COB1A P2COB1AL P2COB P2COB P2COB P2COB3S
PL/1	Work Area 1 (used with all Functions) Work Area 2 for Functions 1 and 1E Regular Work Area 2 for Functions 1A, BL, BN Long Work Area 2 for Functions 1A and BL Work Area 2 for Function 2 Work Area 2 for Function 3 Work Area 2 for Function 3C Work Area 2 for Function 3S	P1PL1 P2PL1 P2PL11A P2PL11AL P2PL1 P2PL1 P2PL1 P2PL13S
NATURAL	Work Area 1 (used with all Functions) Work Area 2 for Functions 1 and 1E Regular Work Area 2 for Functions 1A, BL, BN Long Work Area 2 for Functions 1A and BL Work Area 2 for Function 2 Work Area 2 for Function 3 Work Area 2 for Function 3C Work Area 2 for Function 3S	GEOLP1 GEOLP2 GEOLP21A GEOLP2AL GEOLP22 GEOLP23 GEOLP2 GEOLP23S

Language	Work Area	Copy File
BAL	Work Area 1 (used with all Functions) Work Area 2 for Functions 1 and 1E Regular Work Area 2 for Functions 1A, BL, BN Long Work Area 2 for Functions 1A and BL Work Area 2 for Function 2 Work Area 2 for Function 3 Work Area 2 for Function 3C Work Area 2 for Function 3S	P1BAL P2BAL P2BAL1A P2BAL1A P2BAL P2BAL P2BAL P2BAL3S
C	All Work Areas for all Functions	PAC

APPENDIX 13: CHARACTER-ONLY WORK AREA LAYOUTS (COW)

(as of Geosupport System Software Version 10.1)

This appendix contains layouts of all of the work areas used with the Geosupport System's API. These layouts are current as of the Geosupport software version indicated above. The layouts are in the Character-Only Work Area (COW) format[4].

Some Geosupport functions can only be called using one work area, Work Area 1 (WA1). Other functions can be called using two work areas, WA1 and Work Area 2 (WA2). WA1 contains both input fields (fields used to pass data from the application to Geosupport) and output fields (fields used to pass data from Geosupport to the application). WA1 is organized so that all the input fields occur first, followed by all the output fields. WA2 contains output fields only.

All functions use the same WA1 layout, but the set of WA1 fields that are used depends on the function. In the layout of WA1 in this appendix, the column labeled 'Functions' indicates which functions use each field.

The functions that can be called using two work areas use various WA2 layouts of various lengths. In some cases, several functions share a single WA2 layout. For functions 1A and BL, the user has a choice of two WA2 layouts, a 'regular' WA2 and a 'long' WA2.

The following is a list of all of the Geosupport work areas, indicating the length of each in bytes. Functions that are listed together share a single Work Area 2 layout.

Work Area	Length
WA1, all functions	1200
WA2, Function 1	300
Regular WA2, Functions 1A, BL, BN	1,363
Long WA2, Functions 1A and BL	17,750
WA2, Function 1E	300
WA2, Function 2	200
WA2, Function 3	450
WA2, Function 3C	300
WA2, Function 3S	19,274

The majority of the COW fields are identical to the MSW (Mainframe-Specific Work Area) fields. Appendix 3 consists of a data item dictionary describing the fields that occur in the work areas.

Appendix 12 describes the differences between the COWs and MSWs.

---

[4]The mainframe version of Geosupport supports both the Character-Only Work Area (COW) format and the Mainframe-Specific Work Area (MSW (a.k.a. MFS)) format. The layout of the MSWs is in Appendix 2.

Work Area 1 (COW) - All Functions

Field	Size	Positions		Functions[5]
INPUT Fields:				
Geosupport Function Code	2	1	2	All
House Number - Display Format (HND)	16	3	18	1, 1A, 1E
House Number - Sort Format (HNS)	11	19	29	1, 1A, 1E, D*
Low House Number - Display Format (HND)	16	30	45	Internal Use
Low House Number - Sort Format (HNS)	11	46	56	D*, Internal Use
B10SC-1 (includes B5SC-1 and B7SC-1)	11	57	67	All but BL & BN
Borough Code-1	1	57	57	All but BL & BN
10SC[6]-1	10	58	67	All but BL & BN
Street Name-1	32	68	99	All but BL, BN,D*
B10SC-2 (includes B5SC-2 and B7SC-2)	11	100	110	All but 1* & B*
Borough Code[7]-2	1	100	100	All but 1* & B*
10SC-2	10	101	110	All but 1* & B*
Street Name-2	32	111	142	All but 1* & B*
B10SC-3 (includes B5SC-3 and B7SC-3)	11	143	153	D * 3*
Borough Code-3	1	143	143	D*, 3*
10SC-3	10	144	153	D*, 3*
Street Name-3	32	154	185	D*, 3*
BOROUGH BLOCK LOT (BBL)	11	186	196	BL
Borough Code	1	186	186	BL
Tax Block	5	187	191	BL
Tax Lot	4	192	195	BL
Filler for Tax Lot Version Number	1	196	196	*Not Implemented*
Building Identification Number (BIN)	7	197	203	BN
Compass Direction	1	204	204	2, 3C, 3S

[5]**Note**: An asterisk in the second position of a function code is used as a shorthand notation to represent all function codes having the indicated value in the first position, as follows:

1* = 1, 1A, 1E, 1N
3* = 3, 3C, 3S
B* = BL, BN
D* = D, DG, DN

[6]The user may supply either a 5-Digit, 7-Digit or 10-Digit Street code in this field. The contents are to be left-justified and blank-filled.

[7]The second and third borough codes are only required if they differ from the first.

Work Area 1 (COW) - All Functions (continued)

Field	Size	Positions		Functions
Compass Direction for 2nd Intersection	1	205	205	3S
Filler	7	206	212	
Work Area Format Indicator[8]	1	213	213	All
Filler	101	214	314	
Long Work Area 2 Flag[9]	1	315	315	1A, BL
House Number Justification Flag[10]	1	316	316	*Not Implemented*
House Number Normalization Length[11]	2	317	318	*Not Implemented*
House Number Normalization Override Flag	1	319	319	Internal Use
Street Name Normalization Length Limit	2	320	321	All
Street Name Normalization Format Flag[12]	1	322	322	All
Cross Street Names Flag[13]	1	323	323	1, 1E, 2, 3, 3C
Roadbed Request Switch	1	324	324	1, 1E
Filler	36	325	360	

OUTPUT Fields:

Field	Size	Positions		Functions
First Borough Name	9	361	369	All but D*
House Number - Display Format	16	370	385	1, 1A, 1E, D*
House Number - Sort Format	11	386	396	1, 1A, 1E
B10SC - First Borough and Street Code	11	397	407	1*, 2, 3*
First Street Name Normalized	32	408	439	All but B*

---

[8]When the Work Area Format Indicator (a.k.a. the Platform Indicator) is set to C, Character-Only formats of the work areas (i.e., COW, the formats documented in this appendix) are used. A blank in this indicator means that the mainframe-specific work areas, known as MSW, are used.

[9]The Long Work Area 2 Flag is set to L to request the Long Work Area 2. At present it may only be set to L for Functions 1A and BL and means that a list of BINS will be returned in Work Area 2 to the user in place of the list of addresses.

[10]If the house number is to be right justified, the House Number Justification Flag is set to R and if the house number is to be left justified, the House Number Justification Flag is set to L or left blank.

[11]The House Number Normalization Length field is used to achieve compatibility between the Mainframe-Specific Work Areas (MSW) and the COWs. In the COWs, the House Number is permitted to be 16 characters, but, in the MSW, it is limited to 12 characters. It is not anticipated that users will make use of this field.

[12]If the Street Name Normalization Format Flag is set to S or blank, then the street name is returned in sort format. If it is set to C, then the street name is returned in compact format.

[13]The Cross Street Names Flag (a.k.a Expanded Format Flag), if set to E, will return the street names in the List of Street Names in the output section of Work Area 1. BBL and BIN are also returned where possible.

Work Area 1 (COW) - All Functions **(continued)**

Field	Size	Positions		Functions
B10SC - Second Borough and Street Code	11	440	450	2, 3*
Second Street Name Normalized	32	451	482	2, 3*, D*
B10SC - Third Borough and Street Code	11	483	493	3*
Third Street Name Normalized	32	494	525	3*, D*
BOROUGH BLOCK LOT (BBL)	10	526	535	BL
Borough Code	1	526	526	BL
Tax Block	5	527	531	BL
Tax Lot	4	532	535	BL
Filler for Tax Lot Version Number	1	536	536	*Not Implemented*
Low House Number - Display Format	16	537	552	Internal Use
Low House Number - Sort Format	11	553	563	Internal Use
Building Identification Number	7	564	570	1, 1E, BN
Attribute Bytes - Internal Use Only	3	571	573	Internal Use
Filler	132	574	705	
NIN[14]	6	706	711	*Not Implemented*
Street Attribute Indicator	1	712	712	Internal Use
Reason Code	1	713	713	All
Reason Code Qualifier	1	714	714	*Not Implemented*
Filler	2	715	716	
Geosupport Return Code	2	717	718	All
Message	80	719	798	All
Number of Street Codes and Names in List (up to 10)	2	799	800	1*,2,3*
10 B7SC's	80	801	880	1*,2,3*
List of Street Names (10 Street Name Fields, 32 Bytes Each)	320	881	1200	1*,2,3*

[14]NAP Identification Number

Work Area 2 (COW) - Functions 1 and 1E

Blockface Defined by Address Range Along a Street

Field	Size	Positions		Comments
Internal Use	21	1	21	
Continuous Parity Indicator/Duplicate Address Indicator	1	22	22	
Low House Number of Block face-Sort Format	11	23	33	
High House Number of Block face-Sort Format	11	34	44	
DCP Preferred LGC[15]	2	45	46	
Number of Cross Streets at Low Address End	1	47	47	
List of Cross Streets at Low Address End (Up to 5 B5SCs)	30	48	77	B5SC - Blank-Filled
Number of Cross Streets at High Address End	1	78	78	
List of Cross Streets at High Address End (Up to 5 B5SCs)	30	79	108	B5SC - Blank-Filled
LION KEY	10	109	118	
Borough Code	1	109	109	
Face Code	4	110	113	
Sequence Number	5	114	118	
Special Address Generated Record Flag	1	119	119	
Side of Street Indicator	1	120	120	
Segment Length in Feet	5	121	125	
Spatial Coordinates of Address:				
X Coordinate	7	126	132	
Y Coordinate	7	133	139	
Reserved for Possible Z Coordinate	7	140	146	
Interim Assistance Eligibility Indicator	1	147	147	
Also known as Community Development Eligibility Indicator				
Marble Hill/Rikers Island Alternative Borough Flag	1	148	148	
DOT Street Light Contractor Area	1	149	149	
Community District:	3	150	152	
Community District Borough Code	1	150	150	
Community District Number	2	151	152	
Zip Code	5	153	157	
Function 1E Items	14	158	171	**Use ONLY for Function 1E**
Election District	3	158	160	Invalid for Fn 1
Assembly District	2	161	162	Invalid for Fn 1

[15] For Function 1E, the Board of Elections preferred LGC is provided.

Work Area 2 (COW) - Functions 1 and 1E (continued)

Blockface Defined by Address Range Along a Street

Field	Size	Positions		Comments
Split Election District Flag	1	163	163	Invalid for Fn 1
Congressional District	2	164	165	Invalid for Fn 1
State Senatorial District	2	166	167	Invalid for Fn 1
Civil Court District	2	168	169	Invalid for Fn 1
City Council District	2	170	171	Invalid for Fn 1
Health Center District	2	172	173	
Health Area	4	174	177	
Sanitation District	3	178	180	
Sanitation Collection Scheduling Section and Subsection	2	181	182	
Sanitation Regular Collection Schedule	5	183	187	
Sanitation Recycling Collection Schedule	3	188	190	
Police Patrol Borough Command	1	191	191	
Police Precinct	3	192	194	
Fire Division	2	195	196	
Fire Battalion	2	197	198	
Fire Company Type	1	199	199	
Fire Company Number	3	200	202	
Split Community School District Flag	1	203	203	
Community School District	2	204	205	
Dynamic Block	3	206	208	
Instructional Region	2	209	210	
Feature Type Code	1	211	211	
Segment Type Code	1	212	212	
Alley/Cross Streets flag	1	213	213	
Filler	4	214	217	
1990 Census Tract	6	218	223	
2000 Census Tract	6	224	229	
2000 Census Block	4	230	233	
Filler	40	234	273	
Underlying HNS on True Street	11	274	284	
Underlying B7SC	8	285	292	
Segment Identifier	7	293	299	
Curve Flag	1	300	300	

Work Area 2 (COW) - Functions 1A, BL and BN

Property Defined by Address, BBL, or BIN

Field	Size	Position		Comment
Internal Use	21	1	21	
Continuous Parity Indicator /Duplicate Address Indicator	1	22	22	
Low House Number of Defining Address Range	11	23	33	Sort Format
Borough-Tax Block-Tax Lot (BBL):	10	34	44	Billing BBL if Condo
Borough Code	1	34	34	
Tax Block	5	35	39	
Tax Lot	4	40	43	
Filler for Tax Lot Version Number	1	44	44	*Not Implemented*
RPAD Self-Check Code (SCC) for BBL	1	45	45	
Filler	1	46	46	
RPAD Building Classification Code	2	47	48	
Corner Code	2	49	50	
Number of Existing Structures on Lot	4	51	54	
Number of Street Frontages of Lot	2	55	56	
Interior Lot Flag	1	57	57	
Vacant Lot Flag	1	58	58	
Irregularly-Shaped Lot Flag	1	59	59	
Marble Hill/Rikers Island Alternate Borough Flag	1	60	60	
List of Geographic Identifiers Overflow Flag	1	61	61	When = 'E', there are more than 21 addresses for Fns 1A and BL.
STROLLING KEY:	19	62	80	*Not Implemented*
Borough	1	62	62	
5-Digit Street Code of 'ON' Street	5	63	67	
Side of Street Indicator	1	68	68	
High House Number	11	69	79	Sort Format
Filler	1	80	80	
Reserved for Internal Use	1	81	81	
Building Identification Number (BIN) of Input Address or NAP	7	82	88	
Condominium Flag	1	89	89	If condo, Flag = C

Work Area 2 (COW) - Functions 1A, BL and BN (continued)

Property Defined by Address, BBL, or BIN

Field	Size	Position		Comment
Filler	1	90	90	
DOF Condominium Identification Number	4	91	94	
Condominium Unit ID Number	7	95	101	*Not Implemented*
Condominium Billing BBL	10	102	111	
Tax Lot Version Number for Billing BBL	1	112	112	*Not Implemented*
Self-Check Code (SCC) of Billing BBL	1	113	113	
Low BBL of this Building's Condominium Units	10	114	123	
Tax Lot Version Number of Low BBL	1	124	124	*Not Implemented*
High BBL of this Building's Condominium Units	10	125	134	
Tax Log Version Number of High BBL	1	135	135	*Not Implemented*
Filler	15	136	150	
Cooperative ID Number	4	151	154	
SBVP (Sanborn Map Identifier):	8	155	162	
Sanborn Borough Code	1	155	155	
Volume Number	2	156	157	
Volume Number Suffix	1	158	158	
Page Number	3	159	161	
Page Number Suffix	1	162	162	
DCP Commercial Study Area	5	163	167	
Tax Map Number Section & Volume	5	168	172	
Reserved for Tax Map Page Number	4	173	176	*Not Implemented*
Multiple BBL Flag	1	177	177	These fields will
Next BBL	11	178	188	be used with Mult
Previous BBL	11	189	199	Entity NAPs
Spatial Coordinates of Internal Label Point:				
X Coordinate	7	200	206	
Y Coordinate	7	207	213	
Filler	25	214	238	
Internal Use	8	239	246	
Number of Entries in List of Geographic Identifiers	4	247	250	

Work Area 2 (COW) - Functions 1A, BL and BN (continued)

Property Defined by Address, BBL, or BIN

Field	Size	Position	Comment
List of Geographic Identifiers:	1,113	251 1,363	
Variable length list of 53-byte entries as follows:			
Low House Number	16		Display format(HND)
High House Number	16		Display format(HND)
Borough Code	1		
5-Digit Street Code	5		
DCP-Preferred Local Group Code (LGC)	2		
Building Identification Number	7		
Side of Street Indicator	1		L - Left, R - Right
Geographic Identifier	1		N - NAP of Simplex G - NAP of Complex X - NAP of part of Complex B - NAUB F - Vacant Frontage W - Blank Wall Q - Pseudo-Address V - Vanity Address R - Real Street Blank - Normal
Filler	4		

Long Work Area 2 (COW) - Functions 1A and BL

Property Defined by Address or BBL

Field	Size	Position		Comment
Same as Regular Work Area 2 - Functions 1A/BL	246	1	246	
Number of Buildings on Tax Lot	4	247	250	
List of Buildings on Tax Lot	17,500	251	17,750	
Variable length list of up to 2,500 entries, each entry is a 7-byte BIN	7			

Work Area 2 (COW) - Function 2

Intersection Defined by Two Intersecting Streets

Field	Size	Position		Comment
Internal Use	21	1	21	
Intersection Replication Counter	1	22	22	
DCP-Preferred LGC for Street 1	2	23	24	
DCP-Preferred LGC for Street 2	2	25	26	
Number of Intersecting Streets	1	27	27	
List of Intersecting Streets	30	28	57	
(Up to five B5SCs, 6 bytes each)				
Compass Direction for Intersection Key or Counter for Multiple Intersections	1	58	58	
Filler	5	59	63	
LION Node Number	7	64	70	
Spatial Coordinates:	21			
X Coordinate	7	71	77	
Y Coordinate	7	78	84	
Filler	7	85	91	
SBVP1 (Sanborn Map Identifier):	8			
Borough Code	1	92	92	
Volume Number	2	93	94	
Volume Number Suffix	1	95	95	
Page Number	3	96	98	
Page Number Suffix	1	99	99	
SBVP2 (Sanborn Map Identifier):	8			
Borough Code	1	100	100	
Volume Number	2	101	102	
Volume Number Suffix	1	103	103	
Page Number	3	104	106	
Page Number Suffix	1	107	107	
Marble Hill/Rikers Island Alternative Borough Flag	1	108	108	
DOT Street Light Contractor Area	1	109	109	
Community District:	3			
Community District Borough Code	1	110	110	
Community District Number	2	111	112	
Zip Code	5	113	117	
Health Area	4	118	121	
Police Patrol Borough Command	1	122	122	
Police Precinct	3	123	125	
Fire Division	2	126	127	
Fire Battalion	2	128	129	

Work Area 2 (COW) - Function 2 (continued)

Intersection Defined by Two Intersecting Streets

Field	Size	Position		Comment
Fire Company Type	1	130	130	
Fire Company Number	3	131	133	
Community School District	2	134	135	
2000 Census Tract	6	136	141	
1990 Census Tract	6	142	147	
List of Pairs of Level Codes	10	148	157	*Not Implemented*
Instructional Region	2	158	159	
Filler	41	160	200	

Work Area 2 (COW) - Function 3

Street Segment Defined by 'ON' Street and Two Cross Streets

Field	Size	Position		Comment
Internal Use	21	1	21	
Duplicate Key Flag or Continuous Parity	1	22	22	
Locational Status of Segment	1	23	23	
County Boundary Indicator	1	24	24	
DCP-Preferred LGC for Street 1	2	25	26	
DCP-Preferred LGC for Street 2	2	27	28	
DCP-Preferred LGC for Street 3	2	29	30	
Number of Cross Streets at Low Address End	1	31	31	
List of Cross Streets at Low Address End (Up to five B5SCs, 6 bytes each)	30	32	61	Blank Filled
Number of Cross Streets at High Address End	1	62	62	
List of Cross Streets at High Address End (Up to five B5SCs, 6 bytes each)	30	63	92	Blank Filled
Cross Street Reversal Flag	1	93	93	
LION KEY	10	94	103	
LION Borough Code	1	94	94	
LION Face Code	4	95	98	
LION Sequence Number	5	99	103	
Generated Record Flag	1	104	104	
Length of Segment in Feet	5	105	109	
Segment Azimuth	3	110	112	
Segment Orientation	1	113	113	
Marble Hill/Rikers Island Alternative Borough Flag	1	114	114	
Filler	19	115	133	
Segment Identifier	7	134	140	
DOT Street Light Contractor Area	1	141	141	
Curve Flag	1	142	142	
Dog Leg Flag	1	143	143	
Feature Type Code	1	144	144	
Segment Type Code	1	145	145	
Filler	5	146	150	

Work Area 2 (COW) - Function 3 (continued)

Street Segment Defined by 'ON' Street and Two Cross Streets

Field	Size	Position		Comment
LEFT SIDE:				
Community District:	3			
Community District Borough Code	1	151	151	
Community District Number	2	152	153	
Low House Number	16	154	169	Display Format
High House Number	16	170	185	Display Format
Reserved for Geosupport Use	32	186	217	
Interim Assistance Eligibility Indicator	1	218	218	
Zip Code	5	219	223	
Health Area	4	224	227	
Police Patrol Borough Command	1	228	228	
Police Precinct	3	229	231	
Fire Division	2	232	233	
Fire Battalion	2	234	235	
Fire Company Type	1	236	236	
Fire Company Number	3	237	239	
Community School District	2	240	241	
Dynamic Block	3	242	244	
Instructional Region	2	245	246	
Filler	7	247	253	
1990 Census Tract	6	252	259	
2000 Census Tract	6	260	265	
2000 Census Block	4	266	269	
Filler	31	270	300	
RIGHT SIDE:				
Community District:	3			
Community District Borough Code	1	301	301	
Community District Number	2	302	303	
Low House Number	16	304	319	Display Format
High House Number	16	320	335	Display Format
Reserved for Geosupport Use	32	336	367	
Interim Assistance Eligibility Indicator	1	368	368	
Zip Code	5	369	373	
Health Area	4	374	377	
Police Patrol Borough Command	1	378	378	

Work Area 2 (COW) - Function 3 (continued)

Street Segment Defined by 'ON' Street and Two Cross Streets

Field	Size	Position		Comment
Police Precinct	3	379	381	
Fire Division	2	382	383	
Fire Battalion	2	384	385	
Fire Company Type	1	386	386	
Fire Company Number	3	387	389	
Community School District	2	390	391	
Dynamic Block	3	392	394	
Instructional Region	2	395	396	
Filler	7	397	403	
1990 Census Tract	6	404	409	
2000 Census Tract	6	410	415	
2000 Census Block	4	416	419	
Filler	31	420	450	

Work Area 2 (COW) - Function 3C

Blockface Defined by 'ON' Street, Two Cross Streets and Compass Direction

Field	Size	Position		Comment
Internal Use	21	1	21	
Duplicate Key Flag or Continuous Parity	1	22	22	
Locational Status of Segment	1	23	23	
County Boundary Indicator	1	24	24	
DCP-Preferred LGC for Street 1	2	25	26	
DCP-Preferred LGC for Street 2	2	27	28	
DCP-Preferred LGC for Street 3	2	29	30	
Number of Cross Streets at Low Address End	1	31	31	
List of Cross Streets at Low Address End	30	32	61	Blank-Filled
(Up to five B5SCs, 6 bytes each)				
Number of Cross Streets at High Address End	1	62	62	
List of Cross Streets at High Address End	30	63	92	Blank-Filled
(Up to five B5SCs, 6 bytes each)				
Cross Street Reversal Flag	1	93	93	
LION KEY	10	94	103	
LION Borough Code	1	94	94	
LION Face Code	4	95	98	
LION Sequence Number	5	99	103	
Generated Record Flag	1	104	104	
Length of Segment in Feet	5	105	109	
Segment Azimuth	3	110	112	
Segment Orientation	1	113	113	
Marble Hill/Rikers Island Alternative Borough				
Flag	1	114	114	
Filler	19	115	133	
Segment Identifier	7	134	140	
DOT Street Light Contractor Area	1	141	141	
Side of Street Indicator	1	142	142	
Curve Flag	1	143	143	
Feature Type Code	1	144	144	
Segment Type Code	1	145	145	
Filler	5	146	150	
Community District:	3			
Community District Borough Code	1	151	151	
Community District Number	2	152	153	
Low House Number of Block Face	16	154	169	Display Format
High House Number of Block Face	16	170	185	Display Format
Alternate Low House Number	16	186	201	Supplied for Continuous
Alternate High House Number	16	202	217	Parity - Display Format

Work Area 2 (COW) - Function 3C (continued)

Blockface Defined by 'ON' Street, Two Cross Streets and Compass Direction

Field	Size	Position		Comment
Interim Assistance Eligibility Indicator	1	218	218	
Zip Code	5	219	223	
Health Area	4	224	227	
Police Patrol Borough Command	1	228	228	
Police Precinct	3	229	231	
Fire Division	2	232	233	
Fire Battalion	2	234	235	
Fire Company Type	1	236	236	
Fire Company Number	3	237	239	
Community School District	2	240	241	
Dynamic Block	3	242	244	
Instructional Region	2	245	246	
Filler	7	247	253	
1990 Census Tract	6	254	259	
2000 Census Tract	6	260	265	
2000 Census Block	4	266	269	
Filler	31	270	300	

Work Area 2 (COW) - Function 3S

Street Stretch Defined by 'ON' Street and Optionally Two Cross Streets

Field	Size	Position		Comment
Internal Use	2	1	2	
Primary/Secondary Street Name Indicator	1	3	3	P = Primary S = Secondary[16]
Borough Code	1	4	4	
5-Digit Street Code of 'on' Street	5	5	9	
LGC	2	10	11	Blank if P in position 3
Filler	10	12	21	Always Blank
NUMBER OF INTERSECTIONS	3	22	24	
LIST OF UP TO 350 Intersections	19,250	25	19,274	
Each List Entry is 55 bytes in length, structured as follows:				
Marble Hill/Rikers Island Flag	1			
Distance from previous intersection in list	5			
Gap Flag	1			
Node Number	7			
Number of streets at this intersection	1			
List of up to 5 Cross Streets as this intersection	40			
Each list entry is 8 bytes in length, Structured as follows:				
Borough Code	1			
5-Digit Street Code	5			
DCP-Preferred LGC	2			

[16]The functionality which creates the street stretches based upon the different LGCs has not been implemented.

APPENDIX 14: GEOSUPPORT COPY FILES (COW)

This appendix contains printouts of the Geosupport COW COPY files for COBOL, Assembler, PL/1, C and NATURAL. (For C, COPY files take the form of header files. For NATURAL, COPY files take the form of Local Data Areas.)

The Geosupport COPY files contain source code layouts of the Geosupport work areas. These files are stored in a COPY library that can be accessed by user application programs at compile time. Each supported programming language has an appropriate declarative statement for referencing COPY files at compile time. The Geosupport COPY files are listed below.

GEOSUPPORT SYSTEM WORK AREA COPY FILES (COW)

COW WORK AREA	FUNCTION(S)	LENGTH(bytes)	- - - - COPY File Name - - - -				
			COBOL	ASSEMBLER	PL/1	C	NATURAL
WA1	All	1,200	P1COB	P1BAL	P1PL1	PAC	GEOLP1
WA2	1, 1E,3C	300	P2COB	P2BAL	P2PL1	PAC	GEOLP2
WA2	2	200	P2COB	P2BAL	P2PL1	PAC	GEOLP22
WA2	3	450	P2COB	P2BAL	P2PL1	PAC	GEOLP23
WA2	1A&BL (regular WA2),BN (*)	1363	P2COB1A	P2BAL1A	P2PL11A	PAC	GEOLP21A
WA2	1A&BL (long WA2) (**)	17,750	P2COB1AL	P2BAL1A	P2PL11AL	PAC	GEOLP2AL
WA2	3S	19,274	P2COB3S	P2BAL3S	P2PL13S	PAC	GEOLP23S

(*) Functions 1A, BL and BN share a single regular WA2 layout.

(**) Functions 1A and BL share a single long WA2 layout. (Function BN does not have the long WA2 option.)

See Section VIII.4 for a detailed discussion of the Geosupport COPY feature.

COBOL COPY Files (COW)

P1COB COPY File

```
*****************************************************************
**** THIS IS THE COBOL STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM ***
**** INDEPENDENT WORK AREA 1.                                   ***
**** COPY FILE - P1COB.                              04/07/98   ***
*****************************************************************
****              INPUT FIELDS                              ****
*****************************************************************
         05  PIWA1-IN-FUNC-CODE              PIC X(2).
         05  GEO-WA1-IN-FUNCTION-CODE REDEFINES PIWA1-IN-FUNC-CODE.
            10  GEO-WA1-IN-FUNCTION-1        PIC X.
            10  GEO-WA1-IN-FUNCTION-2        PIC X.
         05  PIWA1-IN-HOUSENUM-DISPLAY       PIC X(16).
         05  GEO-WA1-IN-HOUSENUM-DISPLAY REDEFINES
             PIWA1-IN-HOUSENUM-DISPLAY.
           10  GEO-WA1-IN-HOUSENUM           PIC X(12).
           10  FILLER                        PIC X(4).
         05  PIWA1-IN-HOUSENUM-SORT          PIC X(11).
         05  PIWA1-IN-LOW-HOUSENUM-DISPLAY   PIC X(16).
         05  GEO-WA1-IN-LO-HOUSENUM-DISPLAY REDEFINES
             PIWA1-IN-LOW-HOUSENUM-DISPLAY.
           10  GEO-WA1-IN-LOW-HOUSENUM       PIC X(12).
           10  FILLER                        PIC X(4).
         05  PIWA1-IN-LOW-HOUSENUM-SORT      PIC X(11).
         05  GEO-WA1-IN-10SC-1.
           10  GEO-WA1-IN-BORO               PIC X.
           10  PIWA1-IN-10SC-1               PIC X(10).
         05  GEO-WA1-IN-STREET-1             PIC X(32).
         05  GEO-WA1-IN-10SC-2.
           10  GEO-WA1-IN-BORO-2             PIC X.
           10  PIWA1-IN-10SC-2               PIC X(10).
         05  GEO-WA1-IN-STREET-2             PIC X(32).
         05  GEO-WA1-IN-10SC-3.
           10  GEO-WA1-IN-BORO-3             PIC X.
           10  PIWA1-IN-10SC-3               PIC X(10).
         05  GEO-WA1-IN-STREET-3             PIC X(32).
         05  GEO-WA1-IN-BBL.
            10  GEO-WA1-IN-BL-BORO           PIC X.
            10  GEO-WA1-IN-BLOCKNUM          PIC X(5).
            10  GEO-WA1-IN-LOTNUM            PIC X(5).
         05  PIWA1-IN-BIN                    PIC X(7).
         05  GEO-WA1-IN-COMPASS.
           10  PIWA1-IN-COMPASS1             PIC X.
           10  PIWA1-IN-COMPASS2             PIC X.
         05  FILLER                          PIC X(7).
         05  GEO-WA1-IN-NON-IBM-MAIN-FRAME   PIC X(1).
         05  FILLER                          PIC X(101).
         05  GEO-WA1-IN-LONG-WORKAREA2-FLAG  PIC X.
         05  PIWA1-IN-HSE-NBR-JUSTIFY        PIC X.
         05  PIWA1-IN-HNL                    PIC X(2).
         05  PIWA1-IN-HSE-NBR-OVER-FLAG      PIC X.
         05  GEO-WA1-IN-SNL                  PIC X(2).
         05  GEO-WA1-IN-COMPACT-NAME-FLAG    PIC X.
         05  GEO-WA1-IN-XSTREET-FLAG         PIC X.
         05  PIWA1-IN-ROADBED-REQ-SWITCH     PIC X.
         05  PIWA1-IN-INTERNAL-USE-LEGACY    PIC X.
         05  FILLER                          PIC X(35).
******************************************************************
*****             OUTPUT FIELDS                             *****
******************************************************************
         05  GEO-WA1-OUT-BORONAME            PIC X(9).
         05  PIWA1-OUT-HOUSENUM-DISPLAY      PIC X(16).
         05  GEO-WA1-OUT-HOUSENUM-DISPLAY REDEFINES
             PIWA1-OUT-HOUSENUM-DISPLAY.
           10  GEO-WA1-OUT-HOUSENUM          PIC X(12).
           10  FILLER                        PIC X(4).
         05  PIWA1-OUT-HOUSENUM-SORT         PIC X(11).
```

P1COB COPY File (continued)

```
05  GEO-WA1-OUT-10SC-1                 PIC X(11).
05  GEO-WA1-OUT-STREET-1               PIC X(32).
05  GEO-WA1-OUT-10SC-2                 PIC X(11).
05  GEO-WA1-OUT-STREET-2               PIC X(32).
05  GEO-WA1-OUT-10SC-3                 PIC X(11).
05  GEO-WA1-OUT-STREET-3               PIC X(32).
05  GEO-WA1-OUT-BBL.
   10  GEO-WA1-OUT-BBL-BORO            PIC X.
   10  GEO-WA1-OUT-BLOCKNUM            PIC X(5).
   10  GEO-WA1-OUT-LOTNUM              PIC X(5).
05  PIWA1-OUT-LOW-HN-DISPLAY           PIC X(16).
05  GEO-WA1-OUT-LOW-HN-DISPLAY REDEFINES
    PIWA1-OUT-LOW-HN-DISPLAY.
  10  GEO-WA1-IN-LOW-HOUSENUM          PIC X(12).
  10  FILLER                           PIC X(4).
05  PIWA1-OUT-LOW-HN-SORT              PIC X(11).
05  GEO-WA1-OUT-BIN                    PIC X(7).
05  GEO-WA1-OUT-STREET-ATTR OCCURS 3 TIMES PIC X.
05  FILLER                             PIC X(138).
05  GEO-WA1-OUT-SND-ATTR               PIC X.
05  GEO-WA1-OUT-REASON-CODE            PIC X.
05  FILLER                             PIC X.
05  GEO-WA1-OUT-WARNING-CODE           PIC XX.
05  GEO-WA1-OUT-RETURN-CODE.
   10  GEO-WA1-OUT-RC-1                PIC X.
   10  GEO-WA1-OUT-RC-2                PIC X.
05  GEO-WA1-OUT-ERROR-MESSAGE          PIC X(80).
05  PIWA1-OUT-NUM-SIMILAR-STRS         PIC X(2).
05  PIWA1-OUT-SIMILAR-B7SC             PIC X(8)
                  OCCURS 10 TIMES.
05  GEO-WA1-OUT-SIMILAR-NAMES          PIC X(32)
                  OCCURS 10 TIMES.
```

P2COB COPY File

```
*****************************************************************
**** THIS IS THE COBOL- STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM**
**** INDEPENDENT WORK AREA 2 FOR FUNCTIONS: 1, 1E, 2, 2C, 3,    **
**** 3C, AND 5.                                                 **
****                                                            **
**** COPY FILE - P2COB.                                         **
**** PLEASE NOTE THAT FUNCTIONS 1 AND 1E SHARE A SINGLE         **
**** WORK AREA 2 LAYOUT, AND FUNCTIONS 2 AND 2C ALSO            **
**** SHARE A SINGLE WORK AREA 2 LAYOUT.              04/03/01   **
*****************************************************************
****             LAST MODIFIED MAY 19, 2006                     **
*****************************************************************

        05  PIWA2                           PIC X(450).

*****************************************************************
****     FOR: FUNCTIONS 1 & 1E      *****************************

        05  PIWA2-FUNCTION1   REDEFINES PIWA2.
            10  GEO-WA2-FN1-ACCESS-KEY          PIC X(21).
            10  GEO-WA2-FN1-CONT-PARITY         PIC X.
            10  PIWA2-FN1-LOW-HOUSENUM-SORT     PIC X(11).
            10  PIWA2-FN1-HI-HOUSENUM-SORT      PIC X(11).
            10  GEO-WA2-FN1-PREFERRED-LGC       PIC X(2).
            10  GEO-WA2-FN1-NUM-X-ST-LOW-END    PIC X.
            10  PIWA2-FN1-LOW-B5SC              PIC X(6)
                                                  OCCURS 5 TIMES.
            10  GEO-WA2-FN1-NUM-X-ST-HI-END       PIC X.
            10  PIWA2-FN1-HI-B5SC               PIC X(6)
                                                  OCCURS 5 TIMES.
            10  PIWA2-FN1-LIONKEY.
                15  PIWA2-FN1-LION-BORO         PIC X.
                15  GEO-WA2-FN1-LIONFACECODE    PIC X(4).
                15  GEO-WA2-FN1-LIONSEQ         PIC X(5).
            10  GEO-WA2-FN1-SPECIAL-ADDR-FLAG   PIC X(1).
            10  PIWA2-FN1-SIDE-OF-STR           PIC X.
            10  GEO-WA2-FN1-SEGMENTLENGTH       PIC X(5).
            10  GEO-WA2-FN1-XCOORD              PIC X(7).
            10  GEO-WA2-FN1-YCOORD              PIC X(7).
            10  FILLER-GSS                      PIC X(8).
            10  GEO-WA2-FN1-MARBLE-RIKER-FLAG   PIC X(1).
            10  GEO-WA2-FN1-SLA                 PIC X.
            10  GEO-WA2-FN1-COMDIST.
                15  GEO-WA2-FN1-COMDIST-BORO    PIC X(1).
                15  GEO-WA2-FN1-COMDIST-NUMBER  PIC X(2).
            10  GEO-WA2-FN1-ZIP                 PIC X(5).

 ***  THE FN1E FIELDS ARE VALID ONLY FOR FUNCTION 1E, NOT FUNC 1.**
            10  GEO-WA2-FN1E-ELECTDIST          PIC X(3).
            10  GEO-WA2-FN1E-ASSEMDIST          PIC X(2).
            10  GEO-WA2-FN1E-SPLIT-ED-FLAG      PIC X(1).
            10  GEO-WA2-FN1E-CONGDIST           PIC X(2).
            10  GEO-WA2-FN1E-SENATEDIST         PIC X(2).
            10  GEO-WA2-FN1E-COURTDIST          PIC X(2).
            10  GEO-WA2-FN1E-COUNCILDIST        PIC X(2).

            10  GEO-WA2-FN1-HCD                 PIC X(2).
            10  GEO-WA2-FN1-HEALTHAREA          PIC X(4).
            10  GEO-WA2-FN1-SANIDIST.
                15  GEO-WA2-FN1-SANIDIST-BORO   PIC X(1).
                15  GEO-WA2-FN1-SANIDIST-NUMBER PIC X(2).
            10  GEO-WA2-FN1-SANITATION-SUBSEC   PIC X(2).
            10  GEO-WA2-FN1-SANI-REG            PIC X(5).
            10  GEO-WA2-FN1-SANI-REC            PIC X(3).
            10  GEO-WA2-FN1-POLICEDIST.
               15 GEO-WA2-FN1-POL-PATR-BORO-CMD PIC X(1).
```

P2COB COPY File (continued)

```
              15 GEO-WA2-FN1-POL-PRECINCT      PIC X(3).                 00670000
      ** NOTE:10  GEO-WA2-FN1-FIRESEC ==> FIRE DIVISION **               00680000
              10  GEO-WA2-FN1-FIRESEC          PIC X(2).                 00690000
              10  GEO-WA2-FN1-FIREBAT          PIC X(2).                 00700000
              10  GEO-WA2-FN1-FIRECO.                                    00710000
                  15  GEO-WA2-FN1-FIRECO-TYPE  PIC X(1).                 00720000
                  15  GEO-WA2-FN1-FIRECO-NUM   PIC X(3).                 00730000
              10  GEO-WA2-FN1-SPLIT-SCHOOL-FLAG PIC X.                   00740000
              10  GEO-WA2-FN1-SCHOOLDIST       PIC X(2).                 00750000
              10  GEO-WA2-FN1-DYN-BLOCK        PIC X(3).                 00760000
              10  GEO-WA2-FN1-INSTRUC-DIV      PIC X(2).                 00770006
              10  GEO-WA2-FN1-FEATURE-TYPE     PIC X.                    00780005
              10  GEO-WA2-FN1-SEGMENT-TYPE     PIC X.                    00790012
              10  GEO-WA2-FN1-ALX              PIC X.                    00790115
              10  FILLER                       PIC X(4).                 00791015
              10  GEO-WA2-FN1-1990-CENSUSTRACT PIC X(6).                 00800005
              10  GEO-WA2-FN1-2000-CENS-TRCT   PIC X(6).                 00810005
              10  GEO-WA2-FN1-2000-CENS-BLK    PIC X(4).                 00820005
              10  FILLER                       PIC X(40).                00830010
              10  GEO-WA2-FN1-TRUE-HNS         PIC X(11).                00850010
              10  GEO-WA2-FN1-TRUE-B7SC        PIC X(8).                 00851010
              10  GEO-WA2-FN1-SEG-ID           PIC X(7).                 00860005
              10  GEO-WA2-FN1-CURVE-FLAG       PIC X(1).                 00870005
                                                                         00890005
   ****************************************************************      00900005
   ****     FOR: FUNCTIONS 2       ***********************************   00910005
                                                                         00920005
           05  PIWA2-FUNCTION2   REDEFINES PIWA2.                        00930005
              10  PIWA2-FN2-ACCESS-KEY         PIC X(21).                00940005
              10  GEO-WA2-FN2-DUPINTERFLAG     PIC X.                    00950005
              10  GEO-WA2-FN2-PREFERRED-LGC1   PIC X(2).                 00960005
              10  GEO-WA2-FN2-PREFERRED-LGC2   PIC X(2).                 00970005
              10  GEO-WA2-FN2-NUM-OF-INTERSECTS PIC X.                   00980005
              10  PIWA2-FN2-INTERSECT-B5SC     PIC X(6)                  00990005
                                               OCCURS 5 TIMES.           01000005
              10  GEO-WA2-FN2-COMPDIR          PIC X.                    01010005
              10  FILLER                       PIC X(5).                 01020005
              10  GEO-WA2-FN2-NODE-NUM         PIC X(7).                 01030005
              10  GEO-WA2-FN2-XCOORD           PIC X(7).                 01040005
              10  GEO-WA2-FN2-YCOORD           PIC X(7).                 01050005
              10  FILLER-GSS                   PIC X(7).                 01060005
              10  GEO-WA2-FN2-SANBORN1-BVOLPAGE.                         01070005
                 15 GEO-WA2-FN2-SANBORN1-BORO     PIC X(1).              01080005
                 15 GEO-WA2-FN2-SANBORN1-VOL-NUM  PIC X(3).              01090005
                 15 GEO-WA2-FN2-SANBORN1-PAGE-NUM PIC X(4).              01100005
              10  GEO-WA2-FN2-SANBORN2-BVOLPAGE.                         01110005
                 15 GEO-WA2-FN2-SANBORN2-BORO     PIC X(1).              01120005
                 15 GEO-WA2-FN2-SANBORN2-VOL-NUM  PIC X(3).              01130005
                 15 GEO-WA2-FN2-SANBORN2-PAGE-NUM PIC X(4).              01140005
              10  GEO-WA2-FN2-MARBLE-RIKER-FLAG PIC X(1).                01150005
              10  GEO-WA2-FN2-SLA                PIC X.                  01160005
              10  GEO-WA2-FN2-COMDIST.                                   01170005
                  15  GEO-WA2-FN2-COMDIST-BORO    PIC X(1).              01180005
                  15  GEO-WA2-FN2-COMDIST-NUMBER  PIC X(2).              01190005
              10  GEO-WA2-FN2-ZIP              PIC X(5).                 01200005
              10  GEO-WA2-FN2-HEALTHAREA       PIC X(4).                 01210005
              10  GEO-WA2-FN2-POLICEDIST.                                01220005
                 15 GEO-WA2-FN2-POL-PATR-BORO-CMD PIC X(1).              01230005
                 15 GEO-WA2-FN2-POL-PRECINCT      PIC X(3).              01240005
      ** NOTE:10  GEO-WA2-FN2-FIRESEC ==> FIRE DIVISION **               01250005
              10  GEO-WA2-FN2-FIRESEC          PIC X(2).                 01260005
              10  GEO-WA2-FN2-FIREBAT          PIC X(2).                 01270005
              10  GEO-WA2-FN2-FIRECO.                                    01280005
                  15  GEO-WA2-FN2-FIRECO-TYPE  PIC X(1).                 01290005
                  15  GEO-WA2-FN2-FIRECO-NUM   PIC X(3).                 01300005
              10  GEO-WA2-FN2-SCHOOLDIST       PIC X(2).                 01310005
              10  GEO-WA2-FN2-2000-CENS-TRCT   PIC X(6).                 01320005
              10  GEO-WA2-FN2-1990-CENSUSTRACT PIC X(6).                 01330005
```

P2COB COPY File (continued)

```
           10  GEO-WA2-FN2-LEVEL-LIST OCCURS 5 TIMES.                  01340005
               15 GEO-WA2-FN2-LEVEL-CODES                              01350005
                  OCCURS 2 TIMES          PIC X.                       01360005
           10  GEO-WA2-FN2-INSTRUC-DIV    PIC X(2).                    01370005
           10  FILLER                     PIC X(41).                   01380005
                                                                       01390005
***************************************************************** 01400005
****     FOR: FUNCTION  3          ****************************** 01410005
                                                                       01420005
        05  PIWA2-FUNCTION3   REDEFINES PIWA2.                         01430005
           10  GEO-WA2-FN3-ACCESS-KEY     PIC X(21).                   01440005
           10  GEO-WA2-FN3-DUP-KEY-FLAG   PIC X.                       01450005
           10  GEO-WA2-FN3-LOCATION-STATUS  PIC X.                     01460005
           10  GEO-WA2-FN3-COUNTY-BOUNDARY  PIC X.                     01470005
           10  GEO-WA2-FN3-PREFERRED-LGC1   PIC X(2).                  01480005
           10  GEO-WA2-FN3-PREFERRED-LGC2   PIC X(2).                  01490005
           10  GEO-WA2-FN3-PREFERRED-LGC3   PIC X(2).                  01500005
           10  GEO-WA2-FN3-NUM-X-ST-LOW-END PIC X.                     01510005
           10  PIWA2-FN3-LOW-B5SC           PIC X(6)                   01520005
                                              OCCURS 5 TIMES.          01530005
           10  GEO-WA2-FN3-NUM-X-ST-HI-END  PIC X.                     01540005
           10  PIWA2-FN3-HI-B5SC            PIC X(6)                   01550005
                                              OCCURS 5 TIMES.          01560005
           10  GEO-WA2-FN3-REVERSALFLAG     PIC X.                     01570005
           10  PIWA2-FN3-LIONKEY.                                      01580005
               15  PIWA2-FN3-LION-BORO      PIC X.                     01590005
               15  GEO-WA2-FN3-LIONFACECODE PIC X(4).                  01600005
               15  GEO-WA2-FN3-LIONSEQ      PIC X(5).                  01610005
           10  GEO-WA2-FN3-GENRECFLAG       PIC X.                     01620005
           10  PIWA2-FN3-SEG-LEN            PIC X(5).                  01630005
           10  GEO-WA2-FN3-SEGMENTSLOPE     PIC X(3).                  01640005
           10  GEO-WA2-FN3-SEGMENTORIENT    PIC X.                     01650005
           10  GEO-WA2-FN3-MARBLE-RIKER-FLAG  PIC X(1).                01660005
           10  FILLER                       PIC X(19).                 01670005
           10  GEO-WA2-FN3-SEG-ID           PIC X(7).                  01680005
           10  GEO-WA2-FN3-SLA              PIC X.                     01690005
           10  GEO-WA2-FN3-CURVE-FLAG       PIC X.                     01700005
           10  GEO-WA2-FN3-DOG-LEG          PIC X.                     01710005
           10  GEO-WA2-FN3-FEATURE-TYPE     PIC X.                     01720005
           10  GEO-WA2-FN3-SEGMENT-TYPE     PIC X.                     01721013
           10  FILLER                       PIC X(5).                  01730013
           10  PIWA2-FN3-LEFT-SIDE-OF-STR.                             01740005
               15  GEO-WA2-FN3-LEFT-COMDIST.                           01750005
                 20  GEO-WA2-FN3-LEFT-COMDIST-BORO PIC X(1).           01760005
                 20  GEO-WA2-FN3-LEFT-COMDIST-NUM  PIC X(2).           01770005
               15  PIWA2-FN3-L-LOW-HOUSENUM        PIC X(16).          01780005
               15  PIWA2-FN3-L-HI-HOUSENUM         PIC X(16).          01790005
               15  FILLER-GSS                      PIC X(33).          01800005
               15  GEO-WA2-FN3-LEFT-ZIP            PIC X(5).           01810005
               15  GEO-WA2-FN3-LEFT-HEALTHAREA     PIC X(4).           01820005
               15  GEO-WA2-FN3-LEFT-POLDIST.                           01830005
                20  GEO-WA2-FN3-L-POL-PATR-BOR-CMD PIC X(1).           01840005
                20  GEO-WA2-FN3-L-POL-PRECINCT     PIC X(3).           01850005
   ** NOTE:10  GEO-WA2-3L-R-FIRESEC ==> FIRE DIV **                    01860005
               15  GEO-WA2-3L-L-FIRESEC            PIC X(2).           01870005
               15  GEO-WA2-3L-L-FIREBAT            PIC X(2).           01880005
               15  GEO-WA2-3L-L-FIRECO.                                01890005
                   20  GEO-WA2-3L-L-FIRECO-TYPE    PIC X(1).           01900005
                   20  GEO-WA2-3L-L-FIRECO-NUM     PIC X(3).           01910005
               15  GEO-WA2-FN3-LEFT-SCHLDIST       PIC X(2).           01920005
               15  GEO-WA2-3L-L-DYN-BLOCK          PIC X(3).           01930005
               15  GEO-WA2-3L-L-INSTRUC-DIV        PIC X(2).           01940006
               15  FILLER                          PIC X(7).           01950005
               15  GEO-WA2-3L-L-1990-CENSUSTRACT   PIC X(6) .          01960005
               15  GEO-WA2-3L-L-2000-CENS-TRCT     PIC X(6).           01970005
               15  GEO-WA2-3L-L-2000-CENS-BLK      PIC X(4).           01980005
               15  FILLER                          PIC X.              01990005
               15  FILLER                          PIC X(30).          02000005
```

P2COB COPY File (continued)

```
           10  PIWA2-FN3-RIGHT-SIDE-OF-STR.
               15  GEO-WA2-FN3-RIGHT-COMDIST.
                 20 GEO-WA2-FN3-RIGHT-COMDIST-BORO PIC X(1).
                 20 GEO-WA2-FN3-RIGHT-COMDIST-NUM  PIC X(2).
               15  PIWA2-FN3-R-LOW-HOUSENUM        PIC X(16).
               15  PIWA2-FN3-R-HI-HOUSENUM         PIC X(16).
               15  FILLER-GSS                      PIC X(33).
               15  GEO-WA2-FN3-RIGHT-ZIP           PIC X(5).
               15  GEO-WA2-FN3-RIGHT-HEALTHAREA    PIC X(4).
               15  GEO-WA2-FN3-RIGHT-POLDIST.
                20 GEO-WA2-FN3-R-POL-PATR-BOR-CMD  PIC X(1).
                20 GEO-WA2-FN3-R-POL-PRECINCT      PIC X(3).
               15  GEO-WA2-3L-R-FIRESEC            PIC X(2).
               15  GEO-WA2-3L-R-FIREBAT            PIC X(2).
               15  GEO-WA2-3L-R-FIRECO.
                   20  GEO-WA2-3L-R-FIRECO-TYPE    PIC X(1).
                   20  GEO-WA2-3L-R-FIRECO-NUM     PIC X(3).
               15  GEO-WA2-FN3-RIGHT-SCHLDIST      PIC X(2).
               15  GEO-WA2-3L-R-DYN-BLOCK          PIC X(3).
               15  GEO-WA2-3L-R-INSTRUC-DIV        PIC X(2).
               15  FILLER                          PIC X(7).
               15  GEO-WA2-3L-R-1990-CENSUSTRACT   PIC X(6).
               15  GEO-WA2-3L-R-2000-CENS-TRCT     PIC X(6).
               15  GEO-WA2-3L-R-2000-CENS-BLK      PIC X(4).
               15  FILLER                          PIC X(31).

*****************************************************************
****     FOR: FUNCTION  3C          ******************************

        05  PIWA2-FUNCTION3C  REDEFINES PIWA2.
           10  GEO-WA2-FN3C-ACCESS-KEY          PIC X(21).
           10  PIWA2-FN3C-DUP-KEY-FLAG          PIC X.
           10  GEO-WA2-FN3C-LOCATION-STATUS     PIC X.
           10  GEO-WA2-FN3C-COUNTY-BOUNDARY     PIC X.
           10  GEO-WA2-FN3C-PREFERRED-LGC1      PIC X(2).
           10  GEO-WA2-FN3C-PREFERRED-LGC2      PIC X(2).
           10  GEO-WA2-FN3C-PREFERRED-LGC3      PIC X(2).
           10  GEO-WA2-FN3C-NUM-X-ST-LOW-END    PIC X.
           10  PIWA2-FN3C-LOW-B5SC              PIC X(6)
                                                 OCCURS 5 TIMES.
           10  GEO-WA2-FN3C-NUM-X-ST-HI-END     PIC X.
           10  PIWA2-FN3C-HI-B5SC               PIC X(6)
                                                 OCCURS 5 TIMES.
           10  GEO-WA2-FN3C-REVERSALFLAG        PIC X.
           10  PIWA2-FN3C-LIONKEY.
               15  PIWA2-FN3C-LION-BORO         PIC X.
               15  GEO-WA2-FN3C-LIONFACECODE    PIC X(4).
               15  GEO-WA2-FN3C-LIONSEQ         PIC X(5).
           10  GEO-WA2-FN3C-GENRECFLAG          PIC X.
           10  PIWA2-FN3C-SEG-LEN               PIC X(5).
           10  GEO-WA2-FN3C-SEGMENTSLOPE        PIC X(3).
           10  GEO-WA2-FN3C-SEGMENTORIENT       PIC X.
           10  GEO-WA2-FN3C-MARBLE-RIKER-FLAG   PIC X(1).
           10  FILLER                           PIC X(19).
           10  GEO-WA2-FN3C-SEG-ID              PIC X(7).
           10  GEO-WA2-FN3C-SLA                 PIC X.
           10  PIWA2-FN3C-SIDE-OF-STR           PIC X.
           10  GEO-WA2-FN3C-CURVE-FLAG          PIC X.
           10  GEO-WA2-FN3C-FEATURE-TYPE        PIC X.
           10  GEO-WA2-FN3C-SEGMENT-TYPE        PIC X.
           10  FILLER                           PIC X(5).
           10  PIWA2-FN3C-BLOCKFACE-INFO.
               15  GEO-WA2-FN3C-COMDIST.
                  20  GEO-WA2-FN3C-COMDIST-BORO   PIC X(1).
                  20  GEO-WA2-FN3C-COMDIST-NUMBER PIC X(2).
               15  PIWA2-FN3C-LOW-HOUSENUM        PIC X(16).
               15  PIWA2-FN3C-HI-HOUSENUM         PIC X(16).
               15  PIWA2-FN3C-LOW-HOUSENUM2       PIC X(16).
```

P2COB COPY File (continued)

```
              15  PIWA2-FN3C-HI-HOUSENUM2        PIC X(16).        02680005
              15  FILLER-GSS                     PIC X.            02690005
              15  GEO-WA2-FN3C-ZIP               PIC X(5).         02700005
              15  GEO-WA2-FN3C-HEALTHAREA           PIC X(4).      02710005
              15  GEO-WA2-FN3C-POLICEDIST.                         02720005
               20 GEO-WA2-FN3C-POL-PATR-BORO-CMD PIC X(1).         02730005
               20 GEO-WA2-FN3C-POL-PRECINCT      PIC X(3).         02740005
    ** NOTE:10    GEO-WA2-FN3C-FIRESEC ==> FIRE DIVISION **        02750005
              15  GEO-WA2-FN3C-FIRESEC           PIC X(2).         02760005
              15  GEO-WA2-FN3C-FIREBAT           PIC X(2).         02770005
              15  GEO-WA2-FN3C-FIRECO.                             02780005
                  20  GEO-WA2-FN3C-FIRECO-TYPE   PIC X(1).         02790005
                  20  GEO-WA2-FN3C-FIRECO-NUM    PIC X(3).         02800005
              15  GEO-WA2-FN3C-SCHOOLDIST        PIC X(2).         02810005
              15  GEO-WA2-FN3C-DYN-BLOCK         PIC X(3).         02820005
              15  GEO-WA2-FN3C-INSTRUC-DIV       PIC X(2).         02830006
              15  FILLER                         PIC X(7).         02840005
              15  GEO-WA2-FN3C-1990-CENSUSTRACT  PIC X(6).         02850005
              15  GEO-WA2-FN3C-2000-CENS-TRCT    PIC X(6).         02860005
              15  GEO-WA2-FN3C-2000-CENS-BLK     PIC X(4).         02870005
              15  FILLER                         PIC X(31).        02890009
                                                                   02900005
*******************************************************************02910005
****     FOR: FUNCTION 5           ******************************  02920005
                                                                   02930005
        05  PIWA2-FUNCTION5   REDEFINES PIWA2.                     02940005
           10  GEO-WA2-FN5-ADDR-MATCHING-KEY      PIC X(28).       02950005
```

P2COB1A COPY File

```
******************************************************************  00000100
**  THIS IS THE COBOL STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM **  00000200
**  INDEPENDENT REGULAR WORK AREA 2 FOR FUNCTIONS: 1A, BL, AND **  00000300
**  BN.  THESE THREE FUNCTIONS SHARE A SINGLE WORK AREA 2      **  00000400
**  LAYOUT.  COPY FILE - P2COB1A.                 10/10/97     **  00000500
******************************************************************  00000600
                                                                    00000700
      05  GEO-WA2-1A-ACCESS-KEY            PIC X(21).               00000800
      05  GEO-WA2-1A-CONT-PARITY           PIC X.                   00000900
      05  PIWA2-1A-LOW-HOUSENUM            PIC X(11).               00001000
      05  GEO-WA2-1A-ALTKEY-1.                                      00001100
         10 GEO-WA2-1A-ALTKEY-1-BORO       PIC X.                   00001200
         10 GEO-WA2-1A-ALTKEY-1-TAXBLOCK   PIC X(5).                00001300
         10 GEO-WA2-1A-ALTKEY-1-TAXLOT     PIC X(4).                00001400
      05  FILLER                           PIC X.                   00001500
      05  GEO-WA2-1A-SCC                   PIC X(1).                00001600
      05  FILLER                           PIC X.                   00001700
      05  GEO-WA2-1A-GENERAL-LOT-INFO.                              00001800
         10 GEO-WA2-1A-RPAD-BLDG-CLASS     PIC X(2).                00001900
         10 GEO-WA2-1A-CORNER-CODE         PIC X(2).                00002000
         10 GEO-WA2-1A-TOT-NBR-BLDG        PIC X(4).                00002101
         10 GEO-WA2-1A-NUM-OF-BLOCKFACES   PIC X(2).                00002200
         10 GEO-WA2-1A-INTERIOR-FLAG       PIC X.                   00002300
         10 GEO-WA2-1A-VACANT-FLAG         PIC X.                   00002400
         10 GEO-WA2-1A-IRREG-FLAG          PIC X.                   00002500
      05  GEO-WA2-1A-ALT-BORO-FLAG         PIC X.                   00002600
      05  GEO-WA2-1A-OVERFLOW-FLAG         PIC X(1).                00002700
      05  PIWA2-1A-STROLL-KEY              PIC X(19).               00002800
      05  FILLER-GSS                       PIC X.                   00002900
      05  GEO-WA2-1A-BLDG-ID-NUM           PIC X(7).                00003000
      05  GEO-WA2-1A-CONDO-LOT-FLAG        PIC X.                   00003100
      05  FILLER                           PIC X.                   00003200
      05  GEO-WA2-1A-RPAD-COND-NUM         PIC X(4).                00003300
      05  FILLER                           PIC X(7).                00003400
      05  GEO-WA2-1A-CONDO-BILLING-BBL     PIC X(10).               00003500
      05  FILLER                           PIC X.                   00003600
      05  GEO-WA2-1A-CONDO-BILL-BBL-SCC    PIC X(1).                00003700
      05  GEO-WA2-1A-CONDO-LOW-BBL         PIC X(10).               00003800
      05  FILLER                           PIC X.                   00003900
      05  GEO-WA2-1A-CONDO-HIGH-BBL        PIC X(10).               00004000
      05  FILLER                           PIC X.                   00004100
      05  FILLER                           PIC X(15).               00004200
      05  GEO-WA2-1A-CO-OP-NBR             PIC X(4).                00004300
      05  GEO-WA2-1A-SANBORN-BVOLPAGE.                              00004400
          10  GEO-WA2-1A-SANBORN-BORO      PIC X(1).                00004500
          10  GEO-WA2-1A-SANBORN-VOL-PAGE.                          00004600
            15  GEO-WA2-1A-SANBORN-VOL-NUM   PIC X(3).              00004700
            15  GEO-WA2-1A-SANBORN-PAGE-NUM  PIC X(4).              00004800
      05  GEO-WA2-1A-COMMERC-DIST          PIC X(5).                00004900
      05  GEO-WA2-1A-DOF-MAP-BOROUGH       PIC X.                   00005003
      05  GEO-WA2-1A-TAX-MAP-NBR           PIC X(4).                00005103
      05  FILLER-FOR-TAX-MAP-PAGE          PIC X(4).                00005204
      05  FILLER                           PIC X(23).               00005304
      05  PIWA2-1A-X-COORD                 PIC X(7).                00005404
      05  PIWA2-1A-Y-COORD                 PIC X(7).                00005504
      05  FILLER                           PIC X(25).               00005604
      05  FILLER-GSS                       PIC X(8).                00005704
      05  PIWA2-1A-NUM-OF-ADDR             PIC X(4).                00005804
      05  PIWA2-1A-ADDR-LIST               OCCURS 21 TIMES.         00005904
         10  PIWA2-1A-LIST-LOW-HOUSENUM    PIC X(16).               00006004
         10  PIWA2-1A-LIST-HI-HOUSENUM     PIC X(16).               00006104
         10  PIWA2-1A-LIST-BORO            PIC X.                   00006204
         10  PIWA2-1A-LIST-5SC             PIC X(5).                00006304
         10  PIWA2-1A-LIST-LGC             PIC X(2).                00006404
         10  GEO-WA2-1A-LIST-BIN           PIC X(7).                00006504
         10  GEO-WA2-1A-LIST-SOS           PIC X.                   00006604
```

P2COB1A COPY File (continued)

```
        10  GEO-WA2-1A-ADDR-TYPE          PIC X.                  00006704
        10  FILLER                        PIC X(4).               00006804
```

P2COB1AL COPY File

```
****************************************************************  00000100
**   THIS IS THE COBOL STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM **  00000200
**   INDEPENDENT LONG WORK AREA 2 FOR FUNCTIONS: 1A, AND BL.    **  00000300
**   THESE TWO FUNCTIONS SHARE A SINGLE LONG WORK AREA 2 LAYOU. **  00000400
**   COPY FILE - P2COB1AL.                           11/06/97   **  00000500
****************************************************************  00000600
                                                                   00000700
      05  GEO-WA2-1AL-ACCESS-KEY              PIC X(21).          00000800
      05  GEO-WA2-1AL-CONT-PARITY             PIC X.              00000900
      05  PIWA2-1AL-LOW-HOUSENUM              PIC X(11).          00001000
      05  GEO-WA2-1AL-ALTKEY-1.                                   00001100
         10 GEO-WA2-1AL-ALTKEY-1-BORO         PIC X.              00001200
         10 GEO-WA2-1AL-ALTKEY-1-TAXBLOCK     PIC X(5).           00001300
         10 GEO-WA2-1AL-ALTKEY-1-TAXLOT       PIC X(4).           00001400
      05  FILLER                              PIC X.              00001500
      05  GEO-WA2-1AL-SCC                     PIC X.              00001600
      05  FILLER                              PIC X.              00001700
      05  GEO-WA2-1AL-GENERAL-LOT-INFO.                           00001800
         10 GEO-WA2-1AL-RPAD-BLDG-CLASS       PIC X(2).           00001900
         10 GEO-WA2-1AL-CORNER-CODE           PIC X(2).           00002000
         10 GEO-WA2-1AL-NUM-OF-STRUCTURES     PIC X(2).           00002100
         10 GEO-WA2-1AL-NUM-OF-BLOCKFACES     PIC X(2).           00002200
         10 GEO-WA2-1AL-INTERIOR-FLAG         PIC X.              00002300
         10 GEO-WA2-1AL-VACANT-FLAG           PIC X.              00002400
         10 GEO-WA2-1AL-IRREG-FLAG            PIC X.              00002500
      05  GEO-WA2-1AL-ALT-BORO-FLAG           PIC X.              00002600
      05  FILLER                              PIC X.              00002700
      05  PIWA2-1AL-STROLL-KEY                PIC X(19).          00002800
      05  FILLER-GSS                          PIC X.              00002900
      05  GEO-WA2-1AL-BLDG-ID-NUM             PIC X(7).           00003000
      05  GEO-WA2-1AL-CONDO-LOT-FLAG          PIC X.              00003100
      05  FILLER                              PIC X.              00003200
      05  GEO-WA2-1AL-RPAD-COND-NUM           PIC X(4).           00003300
      05  FILLER                              PIC X(7).           00003400
      05  GEO-WA2-1AL-CONDO-BILLING-BBL       PIC X(10).          00003500
      05  FILLER                              PIC X.              00003600
      05  GEO-WA2-1AL-CONDO-BILL-BBL-SCC      PIC X.              00003700
      05  GEO-WA2-1AL-CONDO-LOW-BBL           PIC X(10).          00003800
      05  FILLER                              PIC X.              00003900
      05  GEO-WA2-1AL-CONDO-HIGH-BBL          PIC X(10).          00004000
      05  FILLER                              PIC X.              00004100
      05  FILLER                              PIC X(15).          00004200
      05  GEO-WA2-1AL-CO-OP-NBR               PIC X(4).           00004300
      05  GEO-WA2-1AL-SANBORN-BVOLPAGE.                           00004400
          10  GEO-WA2-1AL-SANBORN-BORO        PIC X(1).           00004500
          10  GEO-WA2-1AL-SANBORN-VOL-PAGE.                       00004600
            15  GEO-WA2-1AL-SANBORN-VOL-NUM   PIC X(3).           00004700
            15  GEO-WA2-1AL-SANBORN-PAGE-NUM  PIC X(4).           00004800
      05  GEO-WA2-1AL-COMMERC-DIST            PIC X(5).           00004900
      05  PIWA2-1AL-DOF-MAP-BORO              PIC X.              00005000
      05  PIWA2-1AL-DOF-MAP-SECVOL            PIC X(4).           00005100
 *******  PIWA2-1AL-DOF-MAP-PAGE  NOT IMPLEMENTED                 00005200
      05  PIWA2-1AL-DOF-MAP-PAGE              PIC X(4).           00005300
      05  FILLER                              PIC X(23).          00005401
      05  PIWA2-1AL-X-COORD                   PIC X(7).           00005501
      05  PIWA2-1AL-Y-COORD                   PIC X(7).           00005601
      05  FILLER                              PIC X(25).          00005701
      05  FILLER-GSS                          PIC X(8).           00005801
      05  GEO-WA2-1AL-NUM-OF-BINS             PIC X(4).           00005901
      05  GEO-WA2-1AL-BINS                    PIC X(7)            00006001
                                        OCCURS 2500 TIMES.        00007001
```

P2COB3S COPY File

```
*****************************************************************
***  THIS IS THE COBOL STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM **
***  INDEPENDENT WORK AREA 2 FOR FUNCTION 3S.                   **
***  COPY FILE - P2COB3S.                     09/17/97          **
*****************************************************************
       05  PIWA2-3S-ACCESS-KEY.
          10 FILLER-GSS                    PIC X(2).
          10 PIWA2-3S-PORS-STNAME-IND      PIC X.
          10 PIWA2-3S-BORO                 PIC X.
          10 PIWA2-3S-5SC                  PIC X(5).
          10 PIWA2-3S-LGC                  PIC X(2).
          10 FILLER                        PIC X(10).
       05  PIWA2-3S-NUM-OF-INTERSECTS      PIC X(3).
       05  PIWA2-3S-LIST-OFINTERSECTS      OCCURS 350 TIMES.
          10 PIWA2-3S-MARBLE-RIKERS-FLAG   PIC X.
          10 PIWA2-3S-DISTANCE             PIC X(5).
          10 PIWA2-3S-GAP-FLAG             PIC X.
          10 FILLER                        PIC X(7).
          10 PIWA2-3S-NUM-OF-STR           PIC X.
          10 PIWA2-3S-B7SC                 PIC X(8)
                                           OCCURS 5 TIMES.
```

ASSEMBLER COPY Files (COW)

P1BAL COPY File

```
*/*********************************************************************/ 00010000
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE P1BAL,    ***/ 00020000
*/*****  CONTAINING THE Platform Independent LAYOUT OF WORK AREA 1 ***/ 00030000
*/*********************************************************************/ 00040000
*/*****        Last Updated:  24 FEBRUARY    2006                 ***/ 00040116
*/*********************************************************************/ 00041014
P1BAL    DS   0H                                                        00050000
*/****************************                                          00060000
*/*****  INPUT FIELDS  *******                                          00070000
*/****************************                                          00080000
P1IFUNC  DS   0CL2       FUNCTION CODE                                  00090000
P1IFUNC1 DS    CL1       FUNCTION CODE, BYTE 1                          00100000
P1IFUNC2 DS    CL1       FUNCTION CODE, BYTE 2                          00110000
         SPACE                                                          00120000
P1IHSE#  DS    CL16       UNFORMATED HSNUM FOR FUNCTION: 1; 1A; 1E.     00130001
P1IHSE#S DS    CL11       HOUSE NUMBER (SORT FORMAT)                    00140005
*                         The Following two fields are for Fn 5         00150001
P1ILHS#  DS    CL16       UNFORMATED HSNUM                              00160001
P1ILHS#S DS    CL11       HOUSE NUMBER (SORT FORMAT)                    00170005
         SPACE                                                          00180001
P1IBCD1  DS   0CL11      11 Digit Street Code for Street one            00190000
P1IBORO1 DS    CL1       BORO CODE  (1=MN;2=BX;3=BK;4=QN;5=SI)          00200000
P1ICDE1  DS    CL10       STREET CODE FOR STREET ONE                    00210000
P1ISTRT1 DS    CL32       STREET NAME 1                                 00220000
         SPACE                                                          00230000
P1IBCD2  DS   0CL11      11 Digit Street Code for Street two            00240000
P1IBORO2 DS    CL1        BORO CODE OF CROSS ST. 1                      00250000
P1ICDE2  DS    CL10       STREET CODE FOR STREET TWO                    00260000
P1ISTRT2 DS    CL32       STREET NAME 2                                 00270000
         SPACE                                                          00280000
P1IBCD3  DS   0CL11      11 Digit Street Code for Street Three          00290001
P1IBORO3 DS    CL1        BORO CODE OF CROSS ST. 2                      00300000
P1ICDE3  DS    CL10       STREET CODE FOR STREET THREE                  00310000
P1ISTRT3 DS    CL32       STREET NAME 3                                 00320000
         SPACE                                                          00330000
P1IBBL   DS   0CL10      BORO,BLOCK,LOT FOR "BL" FUNCTION               00340002
P1IBLBOR DS    CL1       BORO FOR FUNCTION "BL"                         00350000
P1IBLOCK DS    CL5       TAX BLOCK - FOR  FUNCTION "BL"                 00360000
P1ILOT   DS    CL4       TAX LOT   - FOR  FUNCTION "BL"                 00370000
P1ITLV#  DS    CL1        Tax Lot Version Number (Not Implemented)      00380000
P1IBIN   DS    CL7        BUILDING ID NUMBER                            00390000
P1ICOMP  DS    CL1        COMPASS DIRECTION (TYPES 2, 3C & 3S)          00400010
P1ICOMP2 DS    CL1        COMPASS DIRECTION (TYPE 3S)                   00401010
         DS    CL7        Filler                                        00410010
P1IPLIND DS    CL1        Platform Indicator                            00420003
*                         Blank = St'd Mainframe                        00430000
*                         P = Platform Independent                      00440000
         DS    CL101      Filler                                        00460002
         SPACE                                                          00470000
*/****************************                                          00480000
*/*****      FLAGS     *******                                          00490000
*/****************************                                          00500000
         SPACE                                                          00510000
P1ILONG  DS    CL1        'L' IF LONG WORKAREA 2 FOR FUNC 1A/BL         00520000
P1IJUST  DS    CL1        HOUSE NUMBER JUSTIFICATION FLAG               00520109
P1IHNL   DS    CL2        House Number Length                           00521009
P1IHNBRF DS    CL1        House Number Override Flag - *, $ or blank    00522009
P1ISNL   DS    CL2        LENGTH STREET NAME IS TO BE NORMALIZED TO     00523009
P1ICMPCT DS    CL1        'C' IF STREET NAMES ARE TO BE COMPACTED       00524009
P1IEXPND DS    CL1        EXPANDED FORMAT FLAG                          00530009
P1IRBRQS DS    CL1        ROADBED REQUEST SWITCH                        00550013
P1IRES01 DS    CL1        RESERVED FOR INTERNAL USE                     00581015
         DS    CL35       FILLER                                        00590009
         SPACE                                                          00600000
*/*****************************                                         00610000
```

P1BAL COPY File (continued)

```
*/*****  OUTPUT FIELDS  *******                                          00620000
*/****************************                                          00630000
         SPACE                                                          00640000
P1OBORO  DS    CL9       BORO NAME                                      00650000
P1OHSE#  DS    CL16      HOUSE NUMBER, NORMALIZED, DISPLAY FORMAT       00660000
P1OHSE#S DS    CL11      HOUSE NUMBER (SORT FORMAT)                     00670005
P1OBCD1  DS   0CL11     11 Digit Street Code for Street one             00680000
P1OBORO1 DS    CL1      BORO CODE  (1=MN;2=BX;3=BK;4=QN;5=SI)           00690000
P1OCDE1  DS    CL10      STREET CODE FOR STREET ONE                     00700000
P1OSTRT1 DS    CL32      STREET 1 NAME, NORMALIZED                      00710004
         SPACE                                                          00720000
P1OBCD2  DS   0CL11     11 Digit Street Code for Street two             00730000
P1OBORO2 DS    CL1       BORO CODE OF CROSS ST. 1                       00740000
P1OCDE2  DS    CL10      STREET CODE FOR STREET TWO                     00750000
P1OSTRT2 DS    CL32      STREET 2 NAME, NORMALIZED                      00760004
         SPACE                                                          00770000
P1OBCD3  DS   0CL11     11 Digit Street Code for Street three           00780000
P1OBORO3 DS    CL1       BORO CODE OF street 3                          00790000
P1OCDE3  DS    CL10      STREET CODE FOR STREET THREE                   00800000
P1OSTRT3 DS    CL32      STREET 3 NAME, NORMALIZED                      00810004
         SPACE                                                          00820000
P1OBBL   DS   0CL11     BORO,BLOCK,LOT FOR "BL" FUNCTION                00830000
P1OBLBOR DS    CL1      BORO FOR FUNCTION "BL"                          00840000
P1OBLOCK DS    CL5      TAX BLOCK - FOR  FUNCTION "BL"                  00850000
P1OLOT   DS    CL4      TAX LOT   - FOR  FUNCTION "BL"                  00860000
P1OTLV#  DS    CL1       Tax Lot Version Number (Not Implemented)       00870000
P1OLHSE  DS    CL16      LOW HOUSE NUMBER DISPLAY FORM                  00880006
P1OLHSES DS    CL11      LOW HOUSE NUMBER SORT FORM                     00881006
P1OBIN   DS    CL7       Output Building Identification Number          00882011
P1OATTR3 DS    CL3       Attribute Bytes - Internal Use                 00883011
         SPACE                                                          00890000
         DS    CL132     FILLER                                         00900011
P1ONIN   DS    CL6       NAP IDENTIFICATION NUMBER                      00900107
P1OATTRB DS    CL1       ATTRIBUTE BYTE FROM SND                        00901006
P1OREASN DS    CL1       REASON CODE                                    00910000
         DS    CL1       FILLER                                         00920000
P1OWARNC DS    CL2       Warning Return Code                            00930002
P1ORC    DS    CL2       GeoSupport Return Code                         00940002
P1OERROR DS    CL80      ERROR MESSAGE                                  00950000
P1O#NAME DS    CL2       NUMBER OF STREET NAMES                         00960002
P1OBRWSE DS    CL80      10 B7SC'S                                      00970002
P1ONAMES DS  10CL32      UP TO 10 STREET NAMES                          00980002
P1END    EQU   *                                                        00990000
P1LENGTH EQU   P1END-P1BAL    LENGTH OF P1BAL                           01000000
```

P2BAL COPY File

```
*/*******************************************************************/ 00010000
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE P2BAL,    ***/ 00020000
*/*****  CONTAINING THE LAYOUT OF WORK AREA 2 FOR FUNCTIONS        ***/ 00030000
*/*****  1, 1E, 2, 2C, 3, 3C. PLEASE NOTE THAT FUNCTIONS 2 AND 2C  ***/ 00040000
*/*****  SHARE A SINGLE WORK AREA 2 LAYOUT.                        ***/ 00050000
*/*******************************************************************/ 00060000
*/*****        Last Date Modified -  7 February 2006              ***/ 00060119
*/*******************************************************************/ 00061008
P2BAL    DS   0H                                                        00070000
P2ACCKEY DS    CL21          ACCESS KEY                                 00080000
P2LAYOUT DS   0CL279                                                    00090000
P2F1CPAR DS    CL1           CONTINUOUS PARITY INDICATOR                00100000
P2F1LHNS DS    CL11          LOW HOUSE NUMBER                           00110000
P2F1HHNS DS    CL11          HIGH HOUSE NUMBER                          00120000
P2F1LGC  DS    CL2           DCP Prefered LGC (Fn 1) - BOE (Fn 1E)      00130000
P2F1#STL DS    CL1           NUMBER OF CROSS STREETS AT LOW END         00140000
P2F1CDEL DS    CL30          UP TO FIVEPB5SC'S FOR LOW END              00150000
P2F1#STH DS    CL1           NUMBER OF CROSS STREETS AT HIGH END        00160000
P2F1CDEH DS    CL30          UP TO FIVE B5SC'S FOR HIGH END             00170000
P2F1LBOR DS    CL1           LION BOROUGH CODE                          00180000
P2F1FACE DS    CL4           LION FACE CODE                             00190000
P2F1SEQ  DS    CL5           LION SEQUENCE NUMBER                       00200000
P2F1SPAD DS    CL1           SPECIAL ADDRESS FLAG                       00210000
P2F1SOS  DS    CL1           SIDE OF STREET INDICATOR                   00220000
P2F1SEGL DS    CL5           SEGMENT LEGNTH                             00230000
P2F1XCOR DS    CL7           X COORDINATE                               00240000
P2F1YCOR DS    CL7           Y COORDINATE                               00250000
P2F1ZCOR DS    CL7           Z Coordinate - Not Impl.                   00260000
P2F1RES1 DS    CL1           RESERVED FOR DCP/GSS USE                   00270000
P2F1MHRI DS    CL1           MARBLE HILL/RIKERS ISLAND FLAG             00280000
P2F1SLA  DS    CL1           STREET LIGHT AREA                          00290000
P2F1CD   DS   0CL3           COMMUNITY DISTRICT                         00300000
P2F1CDB  DS    CL1           COMMUNITY DISTRICT BORO                    00310000
P2F1CDN  DS    CL2           COMMUNITY DISTRICT NUMBER                  00320000
P2F1ZIP  DS    CL5           ZIP CODE                                   00330000
P2F1EED  DS    CL3           ELECTION DISTRICT                          00340000
P2F1EAD  DS    CL2           ASSEMBLY DISTRICT                          00350000
P2F1ESED DS    CL1           SPLIT E.D. FLAG                            00360000
*                 Next four fields are valid only for Fn 1E             00370000
P2F1ECON DS    CL2           CONGRESSIONAL DISTRICT                     00380000
P2F1ESEN DS    CL2           SENATORIAL DISTRICT                        00390000
P2F1ECIV DS    CL2           CIVIL COURT DISTRICT                       00400000
P2F1ECOU DS    CL2           CITY COUNCIL DISTRICT                      00410000
*                                                                       00420000
P2F1HCD  DS    CL2           HEALTH CODE DISTRICT                       00430000
P2F1HA   DS    CL4           HEALTH AREA                                00440000
P2F1SAND DS    CL3           SANITATION DISTRICT                        00450000
P2F1SANT DS    CL2           SANITATION DEPT SUBSECTION                 00460000
P2F1SREG DS    CL5           SANITATION REGULAR PICK-UP                 00470000
P2F1SREC DS    CL3           SANITATION RECYCLE PICK-UP                 00480000
P2F1POL  DS   0CL4           POLICE DISTRICT                            00490000
P2F1PBC  DS    CL1           POLICE PATROL BORO COMMAND                 00500000
P2F1POP  DS    CL3           POLICE PRECINCT                            00510000
P2F1FS   DS    CL2           FIRE DIVISION                              00520000
P2F1FB   DS    CL2           FIRE BATTALION                             00530000
P2F1FC   DS   0CL4           FIRE COMPANY                               00540000
P2F1FCT  DS    CL1           FIRE COMPANY TYPE                          00550000
P2F1FCN  DS    CL3           FIRE COMPANY NUMBER                        00560000
P2F1SSCH DS    CL1           SPLIT SCHOOL DISTRICT FLAG                 00570000
P2F1SCH  DS    CL2           SCHOOL DISTRICT                            00580000
P2F1CPB  DS    CL3           DYNAMIC  BLOCK                             00590000
P2F1INSD DS    CL2           Instructional Division                     00591011
P2F1FEAT DS    CL1           Feature Type Code                          00592013
P2F1STC  DS    CL1           SEGMENT TYPE CODE                          00593019
P2F1ALX  DS    CL1           A=Segment split by Alley                   00594020
*                            X=Cross Streets modified                   00595020
```

P2BAL COPY File (continued)

```
         DS    CL4           Filler                                   00600020
P2F1CT90 DS    CL6           1990 CENSUS TRACT                        00630000
P2F1CT00 DS    CL6           2000 CENSUS TRACT                        00631007
P2F1BL00 DS    CL4           2000 CENSUS BLOCK                        00632007
         DS    CL40          FILLER                                   00670015
P2F1UHNS DS    CL11          Underlying HNS                           00670115
P2F1B7SC DS    CL8           "True" Borough 7 Digit Street Code       00670215
P2F1SEGT DS    CL7           Segment Identifier                       00670315
P2F1CURV DS    CL1           Curve Flag                               00671006
P2F1END  EQU   *                                                      00680000
P2F1LEN  EQU   P2F1END-P2BAL       Length of WA 2 for Fn 1            00690000
*                                                                     00700000
********************************************************************  00710000
         ORG   P2LAYOUT      RESET LOCATION COUNTER FOR FUNCTION 2    00720000
********************************************************************  00730000
*                                                                     00740000
P2F2DUPI DS    CL1           DUPLICATE INTERSECT FLAG                 00750000
P2F2LGC1 DS    CL2           STREET 1 PREFERRED LGC                   00760009
P2F2LGC2 DS    CL2           STREET 2 PREFERRED LGC                   00770009
P2F2#INT DS    CL1           NUMBER OF INTERSECTING STREETS           00780000
P2F2CODE DS    CL30          INTERSECTING B5SC'S                      00790000
P2F2CDUP DS    CL1           COMPASS DIRECTION FOR TWO LOWEST         00790105
         DS    CL5           FILLER                                   00790205
P2F2NDNB DS    CL7           LION NODE NUMBER                         00791003
P2F2XCOR DS    CL7           X COORDINATE                             00850000
P2F2YCOR DS    CL7           Y COORDINATE                             00860000
P2F2ZCOR DS    CL7           Z Coordinate - Not Impl.                 00870000
P2F2SVP1 DS   0CL8           FIRST SANBORN BOROUGH, PAGE, VOLUME      00880000
P2F2SB1  DS    CL1           FIRST SANBORN BOROUGH CODE               00890000
P2F2SP1  DS    CL3           FIRST SANBORN PAGE                       00900000
P2F2SV1  DS    CL4           FIRST SANBORN VOLUME                     00910000
P2F2SVP2 DS   0CL8           SECOND SANBORN BOROUGH, PAGE, VOLUME     00920000
P2F2SB2  DS    CL1           SECOND SANBORN BOROUGH CODE              00930000
P2F2SP2  DS    CL3           SECOND SANBORN PAGE                      00940000
P2F2SV2  DS    CL4           SECOND SANBORN VOLUME                    00950000
P2F2MHRI DS    CL1           MARBLE HILL/RIKERS ISLAND FLAG           00960000
P2F2SLA  DS    CL1           STREET LIGHT AREA                        00970000
P2F2CD   DS   0CL3           COMMUNITY DISTRICT                       00980000
P2F2CDB  DS    CL1           COMMUNITY DISTRICT BORO                  00990000
P2F2CDN  DS    CL2           COMMUNITY DISTRICT NUMBER                01000000
P2F2ZIP  DS    CL5           ZIP CODE                                 01010000
P2F2HA   DS    CL4           HEALTH AREA                              01020000
P2F2POL  DS   0CL4           POLICE DISTRICT                          01030000
P2F2PBC  DS    CL1           POLICE PATROL BORO COMMAND               01040000
P2F2POP  DS    CL3           POLICE PRECINCT                          01050000
P2F2FS   DS    CL2           FIRE DIVISION                            01060000
P2F2FB   DS    CL2           FIRE BATTALION                           01070000
P2F2FC   DS   0CL4           FIRE COMPANY                             01080000
P2F2FCT  DS    CL1           FIRE COMPANY TYPE                        01090000
P2F2FCN  DS    CL3           FIRE COMPANY NUMBER                      01100000
P2F2SCH  DS    CL2           SCHOOL DISTRICT                          01110000
P2F2CT00 DS    CL6           2000 CENSUS TRACT                        01120007
P2F2CT90 DS    CL6           1990 CENSUS TRACT                        01130000
P2F2LEVC DS    CL10          Level Codes                              01140011
P2F2INSD DS    CL2           Instructional Division                   01150011
         DS    CL41          FILLER                                   01160011
P2F2END  EQU   *                                                      01170000
P2F2LEN  EQU   P2F2END-P2BAL       Length of WA 2 for Fn 2/2C         01180000
*                                                                     01190000
********************************************************************  01200000
         ORG   P2LAYOUT      RESET LOCATION COUNTER FOR FUNCTION 3    01210000
********************************************************************  01220000
*                                                                     01230000
P2F3DUPF DS   0CL1           DUPLICATE KEY FLAG                       01240000
P2F3PAR  DS    CL1           CONTINUOUS PARITY INDICATOR              01250000
P2F3LST  DS    CL1           Locational Status of Segment             01260000
P2F3CBI  DS    CL1           County Boundary Indicator                01270000
P2F3LGC1 DS    CL2           STREET 1 PREFERRED LGC                   01280009
```

P2BAL COPY File (continued)

```
P2F3LGC2 DS    CL2            STREET 2 PREFERRED LGC                   01290009
P2F3LGC3 DS    CL2            STREET 3 PREFERRED LGC                   01300009
P2F3#STL DS    CL1            NUMBER OF CROSS STREETS AT LOW END       01310000
P2F3CDEL DS    CL30           CROSS STREET B5SC'S AT LOW END           01320000
P2F3#STH DS    CL1            NUMBER OF CROSS STREETS AT HIGH END      01330000
P2F3CDEH DS    CL30           CROSS STREET B5SC'S AT HIGH END          01340000
P2F3REVF DS    CL1            REVERSAL FLAG                            01350000
P2F3KEY  DS   0CL10           LION KEY                                 01360000
P2F3BOR  DS    CL1            LION BOROUGH CODE                        01370000
P2F3FACE DS    CL4            LION FACE CODE                           01380000
P2F3SEQ  DS    CL5            LION SEQUENCE NUMBER                     01390000
P2F3GEN  DS    CL1            GENERATED RECORD FLAG                    01400000
P2F3SEGL DS    CL5            SEGMENT LENGTH IN FEET                   01410000
P2F3SLOP DS    CL3            SEGMENT SLOPE IN DEGREES                 01420000
P2F3ORNT DS    CL1            SEGMENT ORIENTATION                      01430000
P2F3MHRI DS    CL1            MARBLE HILL/RIKERS ISLAND FLAG           01440000
         DS    CL19           Future Use                               01450007
*                                                                      01460000
*                             Apply to both sides of street            01470000
*                                                                      01480000
P2F3SEGT DS    CL7            Segment Identifier                       01481007
P2F3SLA  DS    CL1            STREET LIGHT AREA                        01490000
P2F3CURV DS    CL1            Curve Flag                               01500006
P2F3DGLG DS    CL1            Dog Leg Flag                             01501007
P2F3FEAT DS    CL1            Feature Type Code                        01502013
P2F3STC  DS    CL1            Segment Type Code                        01503019
         DS    CL5            Future Use                               01510019
*                                                                      01520000
*                             Left Side of Street                      01530000
*                                                                      01540000
P2F3CDL  DS   0CL3            LEFT COMMUNITY DISTRICT                  01550000
P2F3CDBL DS    CL1            LEFT COMMUNITY DISTRICT BORO             01560000
P2F3CDNL DS    CL2            LEFT COMMUNITY DISTRICT NUMBER           01570000
P2F3LO#L DS    CL16           LEFT LOW HOUSE NUMBER                    01580000
P2F3HI#L DS    CL16           LEFT HIGH HOUSE NUMBER                   01590000
         DS    CL32                Future Use                          01600000
P2F3RS2L DS    CL1            RESERVED FOR DCP/GSS USE                 01610000
P2F3ZIPL DS    CL5            LEFT ZIP CODE                            01620000
P2F3HAL  DS    CL4            LEFT HEALTH AREA                         01630000
P2F3POLL DS   0CL4            LEFT POLICE DISTRICT                     01640000
P2F3PBCL DS    CL1            LEFT POLICE PATROL BORO COMMAND          01650000
P2F3POPL DS    CL3            LEFT POLICE PRECINCT                     01660000
P2F3FSL  DS    CL2            LEFT FIRE DIVISION                       01670000
P2F3FBL  DS    CL2            LEFT FIRE BATTALION                      01680000
P2F3FCL  DS   0CL4            LEFT FIRE COMPANY                        01690000
P2F3FCTL DS    CL1            LEFT FIRE COMPANY TYPE                   01700000
P2F3FCNL DS    CL3            LEFT FIRE COMPANY NUMBER                 01710000
P2F3SCHL DS    CL2            LEFT SCHOOL DISTRICT                     01720000
P2F3CPBL DS    CL3            Left DYNAMIC  BLOCK                      01730000
P2F3INSL DS    CL2            Left Instructional Division              01731011
         DS    CL7            Filler                                   01740011
P2F3TR9L DS    CL6            Left 1990 CENSUS TRACT                   01780000
P2F3C00L DS    CL6            Left 2000 CENSUS Tract                   01790007
P2F3B00L DS    CL4            Left 2000 CENSUS BLOCK NUMBER            01800007
         DS    CL1            Left Possible Census Block Suffix        01810007
         DS    CL30           Future Use                               01820007
*                                                                      01830000
*                             Right Side of Street                     01840000
*                                                                      01850000
P2F3CDR  DS   0CL3            RIGHT COMMUNITY DISTRICT                 01860000
P2F3CDBR DS    CL1            RIGHT COMMUNITY DISTRICT BORO            01870000
P2F3CDNR DS    CL2            RIGHT COMMUNITY DISTRICT NUMBER          01880000
P2F3LO#R DS    CL16           RIGHT LOW HOUSE NUMBER                   01890000
P2F3HI#R DS    CL16           RIGHT HIGH HOUSE NUMBER                  01900000
         DS    CL32                Future Use                          01910000
P2F3RS2R DS    CL1            RESERVED FOR DCP/GSS USE                 01920000
P2F3ZIPR DS    CL5            RIGHT ZIP CODE                           01930000
P2F3HAR  DS    CL4            RIGHT HEALTH AREA                        01940000
```

P2BAL COPY File (continued)

```
P2F3POLR DS   0CL4           RIGHT POLICE DISTRICT                   01950000
P2F3PBCR DS    CL1           RIGHT POLICE PATROL BORO COMMAND        01960000
P2F3POPR DS    CL3           RIGHT POLICE PRECINCT                   01970000
P2F3FSR  DS    CL2           RIGHT FIRE DIVISION                     01980000
P2F3FBR  DS    CL2           RIGHT FIRE BATTALION                    01990000
P2F3FCR  DS   0CL4           RIGHT FIRE COMPANY                      02000000
P2F3FCTR DS    CL1           RIGHT FIRE COMPANY TYPE                 02010000
P2F3FCNR DS    CL3           RIGHT FIRE COMPANY NUMBER               02020000
P2F3SCHR DS    CL2           RIGHT SCHOOL DISTRICT                   02030000
P2F3CPBR DS    CL3           Right DYNAMIC  BLOCK                    02040000
P2F3INSR DS    CL2           Right Instructional Division            02041011
         DS    CL7           Filler                                  02050011
P2F3TR9R DS    CL6           Right 1990 CENSUS TRACT                 02090000
P2F3C00R DS    CL6           Right 2000 Census Tract                 02091007
P2F3B00R DS    CL4           Right 2000 CENSUS BLOCK                 02100007
         DS    CL1           Right Possible Census Block Suffix      02120007
         DS    CL30          Future Use                              02130007
P2F3END  EQU   *                                                     02140000
P2F3LEN  EQU   P2F3END-P2BAL      Length of WA 2 for Fn 3            02150000
*                                                                    02160000
*********************************************************************02170000
         ORG   P2LAYOUT      RESET LOCATION COUNTER FOR FUNCTION 3C  02180000
*********************************************************************02190000
*                                                                    02200000
P23CDUPF DS   0CL1           DUPLICATE KEY FLAG                      02210000
P23CPAR  DS    CL1           CONTINUOUS PARITY INDICATOR             02220000
P23CLST  DS    CL1           Locational Status of Segment            02230000
P23CCBI  DS    CL1           County Boundary Indicator               02240000
P23CLGC1 DS    CL2           STREET 1 PREFERRED LGC                  02250000
P23CLGC2 DS    CL2           STREET 2 PREFERRED LGC                  02260000
P23CLGC3 DS    CL2           STREET 3 PREFERRED LGC                  02270000
P23C#STL DS    CL1           NUMBER OF CROSS STREETS AT LOW END      02280000
P23CCDEL DS    CL30          CROSS STREET B5SC'S AT LOW END          02290000
P23C#STH DS    CL1           NUMBER OF CROSS STREETS AT HIGH END     02300000
P23CCDEH DS    CL30          CROSS STREET B5SC'S AT HIGH END         02310000
P23CREVF DS    CL1           REVERSAL FLAG                           02320000
P23CKEY  DS   0CL10          LION KEY                                02330000
P23CBOR  DS    CL1           LION BOROUGH CODE                       02340000
P23CFACE DS    CL4           LION FACE CODE                          02350000
P23CSEQ  DS    CL5           LION SEQUENCE NUMBER                    02360000
P23CGEN  DS    CL1           GENERATED RECORD FLAG                   02370000
P23CSEGL DS    CL5           SEGMENT LENGTH IN FEET                  02380000
P23CSLOP DS    CL3           SEGMENT SLOPE IN DEGREES                02390000
P23CORNT DS    CL1           SEGMENT ORIENTATION                     02400000
P23CMHRI DS    CL1           MARBLE HILL/RIKERS ISLAND FLAG          02410000
         DS    CL19          Future Use                              02420007
*                                                                    02430000
*                            Side of Street Information              02440000
*                                                                    02450000
P23CSEGT DS    CL7           Segment Identifier                      02451007
P23CSLA  DS    CL1           STREET LIGHT AREA                       02460000
P23CSOS  DS    CL1           Side of Street Indicator                02470000
P23CCURV DS    CL1           Curve Flag                              02471006
P23CFEAT DS    CL1           Feature Type Code                       02472013
P23CSTC  DS    CL1           Segment Type Code                       02473019
         DS    CL5           Future Use                              02480019
P23CCD   DS   0CL3           COMMUNITY DISTRICT                      02490000
P23CCDB  DS    CL1           COMMUNITY DISTRICT BORO                 02500000
P23CCDN  DS    CL2           COMMUNITY DISTRICT NUMBER               02510000
P23CLO#  DS    CL16          LOW HOUSE NUMBER                        02520000
P23CHI#  DS    CL16          HIGH HOUSE NUMBER                       02530000
P23CHS2L DS    CL16          2ND LOW HSE # - USED IF ODD & EVEN RANGES  02540000
P23CHS2H DS    CL16          2ND HI HSE #    ARE ON SAME SIDE OF STREET 02550000
P23CRES2 DS    CL1           RESERVED FOR DCP/GSS USE                02560000
P23CZIP  DS    CL5           ZIP CODE                                02570000
P23CHAL  DS    CL4           HEALTH AREA                             02580000
P23CPOL  DS   0CL4           POLICE DISTRICT                         02590000
P23CPBC  DS    CL1           POLICE PATROL BORO COMMAND              02600000
```

P2BAL COPY File (continued)

```
P23CPOP  DS    CL3            POLICE PRECINCT                           02610000
P23CFS   DS    CL2            FIRE DIVISION                             02620000
P23CFB   DS    CL2            FIRE BATTALION                            02630000
P23CFC   DS   0CL4            FIRE COMPANY                              02640000
P23CFCT  DS    CL1            FIRE COMPANY TYPE                         02650000
P23CFCN  DS    CL3            FIRE COMPANY NUMBER                       02660000
P23CSCH  DS    CL2            SCHOOL DISTRICT                           02670000
P23CCPB  DS    CL3            DYNAMIC  BLOCK                            02680000
P23CINSD DS    CL2            Instructional Division                    02681011
         DS    CL7            Filler                                    02690011
P23CTR9  DS    CL6            1990 CENSUS TRACT                         02730000
P23CCT00 DS    CL6            2000 Census Tract                         02731007
P23CBL00 DS    CL4            2000 CENSUS BLOCK                         02740007
         DS    CL1            Possible Census Block Suffix              02760007
         DS    CL30           Future Use                                02770007
P23CEND  EQU   *                                                        02780000
P23CLEN  EQU   P23CEND-P2BAL       Length of WA 2 for Fn 3C             02790000
*                                                                       02800000
*********************************************************************** 02810000
         ORG   P2BAL          RESET LOCATION COUNTER FOR FUNCTION 5     02820000
*********************************************************************** 02830000
*                                                                       02840000
P2F5AMK  DS    CL28           ACCESS MATCHING KEY                       02850000
         DS    CL172                                                    02851017
P2F5END  EQU   *                                                        02860000
P2F5LEN  EQU   P2F5END-P2BAL       Length of WA 2 for Fn 5              02870000
         ORG                                                            02880017
```

P2BAL1A COPY File

```
*/*****************************************************************/ 00000100
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE P2BAL1A,  ***/ 00000200
*/*****  CONTAINING THE LAYOUT OF WORK AREA 2 FOR FUNCTION         ***/ 00000300
*/*****  1A, BL And BN WHICH SHARE A SINGLE WORK AREA 2 LAYOUT.    ***/ 00000400
*/*****  The Long Work Area only applies to Functions 1A and BL.   ***/ 00000500
*/*****  The long work area only exists if the Address Overflow    ***/ 00000600
*/*****  Flag has been set on.                                     ***/ 00000700
*/*****************************************************************/ 00000800
*/*****         LAST UPDATE - 12 November 2003                     ***/ 00000908
*/*****************************************************************/ 00001004
P2BAL1A  DS   0H                                                    00001104
         DS    CL21                                                 00001204
P21ACPAR DS    CL1           CONTINUOUS PARITY INDICATOR            00001304
P21AHSEL DS    CL11          LOW HOUSE NUMBER ON BLOCK - HNS Form   00001404
P21AALT1 DS   0CL11          ALTERNATE KEY                          00001504
P21ABOR1 DS    CL1           ALTERNATE KEY - BORO                   00001604
P21ATXB1 DS    CL5           ALTERNATE KEY - TAX BLOCK              00001704
P21ATXL1 DS    CL4           ALTERNATE KEY - TAX LOT                00001804
         DS    CL1              Future Use                          00001904
P21ARSCC DS    CL1           RPAD SCC                               00002004
         DS    CL1           FILLER                                 00002104
P21AGLI  DS   0CL13          GENERAL LOT INFO                       00002205
P21ARBLC DS    CL2           RPAD BUILDING CLASSIFICATION           00002304
P21ACORC DS    CL2           CORNER CODE                            00002404
P21A#STC DS    CL4           TOTAL NUMBER STRUCTURES                00002504
P21A#BFA DS    CL2           TOTAL NUMBER BLOCKFACES                00002604
P21AINTF DS    CL1           INTERIOR LOT FLAG                      00002704
P21AVACF DS    CL1           VACANT LOT FLAG                        00002804
P21AIRLF DS    CL1           IRREGULARLY-SHAPED LOT FLAG            00002904
*                                                                   00003004
P21AABFL DS    CL1           Marble Hill/ Rikers ALTERNATE BORO FLAG 00003104
P21AOVFL DS    CL1           Address Overflow Flag                  00003204
*                                                                   00003304
P21ASTRK DS    CL19          STROLLING KEY                          00003404
*                                                                   00003504
P21ARFIU DS    CL1           RESERVED FOR INTERNAL USE              00003604
P21ABIN  DS    CL7           BUILDING IDENTIFICATION NUMBER (BIN)   00003704
*                           Condo Information                       00003804
P21ACONF DS    CL1           CONDO LOT FLAG                         00003904
         DS    CL1           Filler for Future Use                  00004004
P21ARCO# DS    CL4           RPAD CONDO NUMBER                      00004104
         DS    CL7           Future Use - Condo Unit Number         00004204
P21ACBBL DS    CL11          CONDO BILLING BBL                      00004304
P21ACBBS DS    CL1           CONDO BILLING BBL SCC                  00004404
P21ACLBL DS    CL11          CONDO LOW BBL                          00004504
P21ACHBL DS    CL11          CONDO HIGH BBL                         00004604
         DS    CL15          Filler                                 00004704
P21ACOOP DS    CL4           Co-op Number                           00004804
*                                                                   00004904
P21ASBVP DS    CL8           SANDBORN BOROUGH/VOLUME/PAGE           00005004
*                                                                   00005104
P21ABUSA DS    CL5           BUSINESS AREA                          00005204
P21ATAXM DS    CL5           Tax Map Number - Section and Volume    00005307
         DS    CL4           Reserved for Tax Map Page              00005408
         DS    CL23             FILLER                              00005508
P21AXCO  DS    CL7           X Coordinate of Annotation Point       00005608
P21AYCO  DS    CL7           Y Coordinate of Annotation Point       00005708
         DS    CL25             FILLER                              00005808
         DS    CL8           Internal Use                           00005908
P21A#ADR DS    CL4           TOTAL ADDRESSES FOR LOT                00006004
P21ALIST DS   0CL1113        LIST OF ADDRESSES, MAXIMUM OF 21       00006104
P21ALOW# DS    CL16          LOW HOUSE NUMBER-Display Form          00006204
P21AHI#  DS    CL16          HIGH HOUSE NUMBER-Display Form         00006304
P21ABCDE DS    CL1           Borough Code                           00006404
P21ACODE DS    CL5           STREET CODE                            00006504
P21APLGC DS    CL2           Preferred LGC                          00006604
```

P2BAL1A COPY File (continued)

```
P21ALBIN DS    CL7           BIN                                            00006704
P21ALSOS DS    CL1           Side of Street Indicator                       00006804
P21AATP  DS    CL1           Address Type Flag                              00006904
         DS    CL4           FILLER                                         00007004
*  STORAGE IS RESERVED FOR THE REMAINING 20 ADDRESS STRUCTURES.             00007104
*  EACH STRUCTURE IS IDENTICAL TO THE ONE DEFINED ABOVE.                    00007204
         DS    CL1060        REMAINING ADDRESSES                            00007304
P21ASEND EQU   *                                                            00007404
P21ASLEN EQU   P21ASEND-P2BAL1A    LENGTH OF Short P2BAL1A                  00007504
*                                                                           00007604
*        Long Work Area Overlay                                             00007704
*                                                                           00007804
         ORG   P21A#ADR                                                     00007904
P21A#BIN DS    CL4           Total Nbr of BINs for Lot                      00008004
P21ABINS DS    2500CL7                                                      00008104
P21ALEND EQU   *                                                            00008204
P21ALLEN EQU   P21ALEND-P2BAL1A    Length of Long P2BAL1A                   00009004
```

P2BAL3S COPY File

```
*/*****************************************************************/ 00000100
*/*****  THIS IS GEOSUPPORT INFORMATION SYSTEM COPY FILE P2BAL3S,  ***/ 00000200
*/*****  CONTAINING THE LAYOUT OF WORK AREA 2 FOR FUNCTION 3S.     ***/ 00000300
*/*****************************************************************/ 00000400
*/******         Last Modified - 3 April 2002                     ***/ 00000502
*/*****************************************************************/ 00000602
P2BAL3S  DS   0H                                                     00000700
P23SAKEY DS   0CL21         ACCESS KEY                               00000800
         DS    CL2          Internal Use Only                        00000900
P23SPORS DS    CL1          P=Primary, S=Secondary                   00001000
P23SBORO DS    CL1          Borough Code                             00001100
P23S5SC  DS    CL5          Street Code                              00001200
P23SLGC  DS    CL2          Blank if P in P23SPORS                   00001300
         DS    CL10         Internal use Only                        00001400
P23S#INT DS    CL3          NUMBER OF INTERSECTIONS ON STRETCH       00001500
*                           Up to 350 Intersections                  00001600
*P23SINT  DS    0CL87       INTERSECTION LAYOUT                      00001700
P23SINT  DS    0CL55        INTERSECTION LAYOUT                      00001800
P23SMHRI DS    CL1          Marble Hill / Rikers Island Flag         00001900
P23SDIST DS    CL5          DISTANCE IN FEET FROM PREVIOUS INTERSECT. 00002000
P23SGAPF DS    CL1          GAP FLAG ("G" IF NO SEGMENT CONNECTS THIS 00002100
*                           INTERSECTION TO THE PREVIOUS ONE)        00002200
P23SNODE DS    CL7          Node Number                              00002301
P23S#ST  DS    CL1          Number of Streets intersecting (max 5)   00002400
P23SCDE1 DS    CL8          NUMERICALLY SMALLEST PB5SC               00002500
P23SCDE2 DS    CL8          NUMERICALLY 2ND SMALLEST PB5SC           00002600
P23SCDE3 DS    CL8          Remaining Street Codes in any order      00002700
P23SCDE4 DS    CL8                                                   00002800
P23SCDE5 DS    CL8                                                   00002900
P23SREST DS    CL19195     REMAINING INTERSECTIONS Assuming Max size 00003402
P23SEND  EQU   *                                                     00003500
P23SLEN  EQU   P23SEND-P2BAL3S     LENGTH OF P2BAL3S                 00003600
```

PL/1 COPY Files (COW)

P1PL1 COPY File

```
/*****************************************************************/
/***  THIS IS THE PL/1 STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM  ***/
/***  INDEPENDENT WORK AREA 1.                                   ***/
/***  COPY FILE - P1PL1.                                         ***/
/***                                              04/07/98       ***/
/*****************************************************************/
 DCL PP1 POINTER;
 DCL
   1 P1PL1,
     /***********************************************/
     /*****          INPUT  FIELDS             *****/
     /***********************************************/
     2 PIWA1_IN_FUNCTION_CODE,
       3 PIWA1_IN_FUNCTION_1             CHAR(1),
       3 PIWA1_IN_FUNCTION_2             CHAR(1),
     2 PIWA1_IN_HOUSENUM_DISPLAY         CHAR(16),
     2 PIWA1_IN_HOUSENUM_SORT            CHAR(11),
     2 PIWA1_IN_LOW_HOUSENUM_DISPLAY     CHAR(16),
     2 PIWA1_IN_LOW_HOUSENUM_SORT        CHAR(11),
     2 PIWA1_IN_BORO_1                   CHAR(1),
     2 PIWA1_IN_10SC_1                   CHAR(10),
     2 PIWA1_IN_STREET_1                 CHAR(32),
     2 PIWA1_IN_BORO_2                   CHAR(1),
     2 PIWA1_IN_10SC_2                   CHAR(10),
     2 PIWA1_IN_STREET_2                 CHAR(32),
     2 PIWA1_IN_BORO_3                   CHAR(1),
     2 PIWA1_IN_10SC_3                   CHAR(10),
     2 PIWA1_IN_STREET_3                 CHAR(32),
     2 PIWA1_IN_BBL,
       3 PIWA1_IN_BBL_BORO               CHAR(1),
       3 PIWA1_IN_BLOCK                  CHAR(5),
       3 PIWA1_IN_LOT                    CHAR(4),
       3 PIWA1_IN_LOT_VER                CHAR(1),
     2 PIWA1_IN_BIN                      CHAR(7),
     2 PIWA1_IN_COMPASS                  CHAR(1),
     2 PIWA1_IN_COMPASS2                 CHAR(1),
     2 FILLER_100                        CHAR(7),
     2 PIWA1_IN_PLATFORM_INDICATOR       CHAR(1),
     2 FILLER_200                        CHAR(101),
     2 PIWA1_IN_LONG_WORKAREA2_FLAG      CHAR(1), /*L=LONG */
     2 PIWA1_IN_HSE_NBR_JUSTIFY          CHAR(1),
     2 PIWA1_IN_HNL                      CHAR(2), /* NI */
     2 PIWA1_IN_HSE_OVER_FLAG            CHAR(1),
     2 PIWA1_IN_SNL                      CHAR(2),
     2 PIWA1_IN_SN_NORM_FORMAT           CHAR(1), /*C=COMPACT */
                                                  /*S OR ' '=SORT*/
     2 PIWA1_IN_EXPANDED_FORMAT          CHAR(1),
     2 PIWA1_IN_ROADBED_REQ_SWITCH       CHAR(1),
     2 PIWA1_IN_INTERNAL_USE_LEGACY      CHAR(1),
     2 FILLER_400                        CHAR(35),
     /***********************************************/
     /*****           OUTPUT  FIELDS           *****/
     /***********************************************/
     2 PIWA1_OUT_BORONAME                CHAR(9),
     2 PIWA1_OUT_HOUSENUM_DISPLAY        CHAR(16),
     2 PIWA1_OUT_HOUSENUM_SORT           CHAR(11),
     2 PIWA1_OUT_B10SC_1                 CHAR(11),
     2 PIWA1_OUT_STREET_1                CHAR(32),
     2 PIWA1_OUT_B10SC_2                 CHAR(11),
     2 PIWA1_OUT_STREET_2                CHAR(32),
     2 PIWA1_OUT_B10SC_3                 CHAR(11),
     2 PIWA1_OUT_STREET_3                CHAR(32),
     2 PIWA1_OUT_BBL,
       3 PIWA1_OUT_BBL_BORO              CHAR(1),
       3 PIWA1_OUT_BLOCK                 CHAR(5),
       3 PIWA1_OUT_LOT                   CHAR(4),
```

P1PL1 COPY File (continued)

```
      2 PIWA1_OUT_LOT_VER                 CHAR(1),
      2 PIWA1_OUT_LO_HOUSENUM_DISPLAY     CHAR(16),
      2 PIWA1_OUT_LO_HOUSENUM_SORT        CHAR(11),
      2 PIWA1_OUT_BIN                     CHAR(7),
      2 PIWA1_OUT_STREET_ATTR(3)          CHAR(1),
      2 FILLER_500                        CHAR(138),
      2 PIWA1_OUT_SND_ATTR                CHAR(1),
      2 PIWA1_OUT_REASON_CODE             CHAR(1),
      2 FILLER_600                        CHAR(1),
      2 PIWA1_OUT_WARNING_CODE            CHAR(2),
      2 PIWA1_OUT_RETURN_CODE             CHAR(2),
      2 PIWA1_OUT_ERROR_MESSAGE           CHAR(80),
      2 PIWA1_OUT_NUM_SIMILAR_STRS        CHAR(2),
      2 PIWA1_OUT_SIMILAR_B7SC(10)        CHAR(8),
      2 PIWA1_OUT_SIMILAR_NAMES(10)       CHAR(32);

  DCL PIWA1_IN_FUNC_CODE                  CHAR(2)
               BASED(ADDR(PIWA1_IN_FUNCTION_CODE));

  DCL WORK1PL1   BASED(PP1)               CHAR(1200);
  PP1=ADDR(P1PL1);
```

P2PL1 COPY File

```
 /*****************************************************************/
 /***  THIS IS THE PL/1 STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM  ***/
 /***  INDEPENDENT WORK AREA 2 FOR FUNCTIONS: 1, 1E, 2, 2C, 3,    ***/
 /***  3C, AND 5.                                                 ***/
 /***                                                             ***/
 /***  COPY FILE - P2PL1.                                         ***/
 /***  PLEASE NOTE THAT FUNCTIONS 1 AND 1E SHARE A SINGLE         ***/
 /***  WORK AREA 2 LAYOUT, AND FUNCTIONS 2 AND 2C ALSO            ***/
 /***  SHARE A SINGLE WORK AREA 2 LAYOUT.              12/30/97   ***/
 /*****************************************************************/
DCL PP2 POINTER;
DCL P2PL1           CHAR(450) INIT(' ');


 /*****************************************************************/
 /*******      FOR: FUNCTIONS 1 & 1E     *************************/
DCL
  1 PIWA2_FUNCTION1   BASED(PP2),
    2 PIWA2_FN1_ACCESS_KEY              CHAR(21),
    2 PIWA2_FN1_CONT_PARITY             CHAR(1),/*(OR DUP ADDR IND)*/
    2 PIWA2_FN1_LOW_HOUSENUM            CHAR(11),/* SORT FORMAT */
    2 PIWA2_FN1_HI_HOUSENUM             CHAR(11),/* SORT FORMAT */
    2 PIWA2_FN1_PREF_LGC                CHAR(2),
    2 PIWA2_FN1_NUM_X_ST_LOW_END        CHAR(1),
    2 PIWA2_FN1_LOW_B5SC(5)             CHAR(6),
    2 PIWA2_FN1_NUM_X_ST_HI_END         CHAR(1),
    2 PIWA2_FN1_HI_B5SC(5)              CHAR(6),
    2 PIWA2_FN1_LIONKEY,
      3 PIWA2_FN1_LION_BORO             CHAR(1),
      3 PIWA2_FN1_LION_FACECODE         CHAR(4),
      3 PIWA2_FN1_LION_SEQ              CHAR(5),
    2 PIWA2_FN1_SPECIAL_ADDR_FLAG       CHAR(1),
    2 PIWA2_FN1_SIDE_OF_STR             CHAR(1),
    2 PIWA2_FN1_SEG_LEN                 CHAR(5),
    2 PIWA2_FN1_XCOORD                  CHAR(7),
    2 PIWA2_FN1_YCOORD                  CHAR(7),
    2 FILLER_100                        CHAR(7), /* FOR ZCOORD */
    2 FILLER_200                        CHAR(1), /* FOR GSS USE*/
    2 PIWA2_FN1_MARBLE_RIKERS_FLAG      CHAR(1),
    2 PIWA2_FN1_DOT_SLA                 CHAR(1),
    2 PIWA2_FN1_COM_DIST,
      3 PIWA2_FN1_COM_DIST_BORO         CHAR(1),
      3 PIWA2_FN1_COM_DIST_NUM          CHAR(2),
    2 PIWA2_FN1_ZIP                     CHAR(5),

    2 PIWA2_FN1E_ELECT_DIST             CHAR(3), /*****************/
    2 PIWA2_FN1E_ASSEM_DIST             CHAR(2), /* THE FNIE      */
    2 PIWA2_FN1E_SPLIT_ED_FLAG          CHAR(1), /* FIELDS ARE    */
    2 PIWA2_FN1E_CONG_DIST              CHAR(2), /* VALID ONLY FOR*/
    2 PIWA2_FN1E_SENATE_DIST            CHAR(2), /* FUNCTION 1E,  */
    2 PIWA2_FN1E_COURT_DIST             CHAR(2), /* NOT FUNC 1.   */
    2 PIWA2_FN1E_COUNCIL_DIST           CHAR(2), /*****************/

    2 PIWA2_FN1_HEALTH_CENTER_DIST      CHAR(2),
    2 PIWA2_FN1_HEALTH_AREA             CHAR(4),
    2 PIWA2_FN1_SANI_DIST,
      3 PIWA2_FN1_SANI_DIST_BORO        CHAR(1),
      3 PIWA2_FN1_SANI_DIST_NUM         CHAR(2),
    2 PIWA2_FN1_SANI_SUBSEC             CHAR(2),
    2 PIWA2_FN1_SANI_REG                CHAR(5),
    2 PIWA2_FN1_SANI_REC                CHAR(3),
    2 PIWA2_FN1_POLICE_DIST,
      3  PIWA2_FN1_POL_PAT_BORO_CMD     CHAR(1),
      3  PIWA2_FN1_POL_PRECINCT         CHAR(3),
    2 PIWA2_FN1_FIRE_DIV                CHAR(2),
    2 PIWA2_FN1_FIRE_BAT                CHAR(2),
```

P2PL1 COPY File (continued)

```
     2 PIWA2_FN1_FIRE_CO,
       3 PIWA2_FN1_FIRE_CO_TYPE          CHAR(1),
       3 PIWA2_FN1_FIRE_CO_NUM           CHAR(3),
     2 PIWA2_FN1_SCHL_DIST_SPLIT_FLAG    CHAR(1),
     2 PIWA2_FN1_SCHL_DIST               CHAR(2),
     2 PIWA2_FN1_DYN_BLK                 CHAR(3),
     2 PIWA2_FN1_INSTRUCT_DIV            CHAR(2),
     2 PIWA2_FN1_FEATURE_TYPE            CHAR(1),
     2 PIWA2_FN1_SEGMENT_TYPE            CHAR(1),
     2 PIWA2_FN1_ALX                     CHAR(1),
     2 FILLER_290                        CHAR(4),
     2 PIWA2_FN1_1990_CENS_TRCT          CHAR(6),
     2 PIWA2_FN1_2000_CENSUS_TRACT       CHAR(6),
     2 PIWA2_FN1_2000_CENSUS_BLOCK       CHAR(4),
     2 FILLER_294_RESV_DCP               CHAR(1),
     2 FILLER_300                        CHAR(50),
     2 PIWA2_FN1_REAL_B7SC               CHAR(8),
     2 PIWA2_FN1_SEGMENT_ID              CHAR(7),
     2 PIWA2_FN1_CURVE_FLAG              CHAR(1);

 DCL   PIWA2_FN1_COMDIST                 CHAR(3)
         BASED(ADDR(PIWA2_FN1_COM_DIST));
 DCL   PIWA2_FN1_SANIDIST                CHAR(3)
         BASED(ADDR(PIWA2_FN1_SANI_DIST));
 DCL   PIWA2_FN1_POLDIST                 CHAR(4)
         BASED(ADDR(PIWA2_FN1_POLICE_DIST));


  /*****************************************************************/
  /*******     FOR: FUNCTIONS 2 & 2C     ****************************/
 DCL
   1 PIWA2_FUNCTION2   BASED(PP2),
     2 PIWA2_FN2_ACCESS_KEY              CHAR(21),
     2 PIWA2_FN2_DUP_INTERSECT_FLAG      CHAR(1),
     2 PIWA2_FN2_PREF_LGC1               CHAR(2),
     2 PIWA2_FN2_PREF_LGC2               CHAR(2),
     2 PIWA2_FN2_NUM_OF_INTERSECTS       CHAR(1),
     2 PIWA2_FN2_INTERSECT_B5SC(5)       CHAR(6),
     2 PIWA2_FN2_COMPDIR                 CHAR(1),
     2 FILLER_350                        CHAR(5),
     2 PIWA2_FN2_LIONNODENUM             CHAR(7),
     2 PIWA2_FN2_XCOORD                  CHAR(7),
     2 PIWA2_FN2_YCOORD                  CHAR(7),
     2 FILLER_400                        CHAR(7), /* FOR ZCOORD */
     2 PIWA2_FN2_SANBORN1,
       3 PIWA2_FN2_SANBORN1_BORO         CHAR(1),
       3 PIWA2_FN2_SANBORN1_VOL          CHAR(3),
       3 PIWA2_FN2_SANBORN1_PAGE         CHAR(4),
     2 PIWA2_FN2_SANBORN2,
       3 PIWA2_FN2_SANBORN2_BORO         CHAR(1),
       3 PIWA2_FN2_SANBORN2_VOL          CHAR(3),
       3 PIWA2_FN2_SANBORN2_PAGE         CHAR(4),
     2 PIWA2_FN2_MARBLE_RIKERS_FLAG      CHAR(1),
     2 PIWA2_FN2_DOT_SLA                 CHAR(1),
     2 PIWA2_FN2_COM_DIST,
       3 PIWA2_FN2_COM_DIST_BORO         CHAR(1),
       3 PIWA2_FN2_COM_DIST_NUM          CHAR(2),
     2 PIWA2_FN2_ZIP                     CHAR(5),
     2 PIWA2_FN2_HEALTH_AREA             CHAR(4),
     2 PIWA2_FN2_POLICE_DIST,
       3  PIWA2_FN2_POL_PAT_BORO_CMD     CHAR(1),
       3  PIWA2_FN2_POL_PRECINCT         CHAR(3),
     2 PIWA2_FN2_FIRE_DIV                CHAR(2),
     2 PIWA2_FN2_FIRE_BAT                CHAR(2),
     2 PIWA2_FN2_FIRE_CO,
       3 PIWA2_FN2_FIRE_CO_TYPE          CHAR(1),
       3 PIWA2_FN2_FIRE_CO_NUM           CHAR(3),
     2 PIWA2_FN2_SCHL_DIST               CHAR(2),
```

P2PL1 COPY File (continued)

```
     2 PIWA2_FN2_2000_CENSUS_TRACT        CHAR(6),
     2 PIWA2_FN2_1990_CENS_TRCT           CHAR(6),
     2 PIWA2_FN2_LEVEL_CODES(5,2)         CHAR(1),
     2 PIWA2_FN2_INSTRUCT_DIV             CHAR(2),
     2 FILLER_500                         CHAR(41);

 DCL   PIWA2_FN2_COMDIST                  CHAR(3)
         BASED(ADDR(PIWA2_FN2_COM_DIST));
 DCL   PIWA2_FN2_POLDIST                  CHAR(4)
         BASED(ADDR(PIWA2_FN2_POLICE_DIST));
 DCL   PIWA2_FN2_SANBORN1_BVOLPAGE        CHAR(8)
         BASED(ADDR(PIWA2_FN2_SANBORN1)),
       PIWA2_FN2_SANBORN2_BVOLPAGE        CHAR(8)
         BASED(ADDR(PIWA2_FN2_SANBORN2));


  /*****************************************************************/
  /*******     FOR: FUNCTION 3         *****************************/
 DCL
   1 PIWA2_FUNCTION3   BASED(PP2),
     2 PIWA2_FN3_ACCESS_KEY               CHAR(21),
     2 PIWA2_FN3_DUP_KEY_FLAG             CHAR(1),/*(OR CONT PARITY)*/
     2 PIWA2_FN3_LOCATION_STATUS          CHAR(1),
     2 PIWA2_FN3_COUNTY_BOUNDARY          CHAR(1),
     2 PIWA2_FN3_PREF_LGC1                CHAR(2),
     2 PIWA2_FN3_PREF_LGC2                CHAR(2),
     2 PIWA2_FN3_PREF_LGC3                CHAR(2),
     2 PIWA2_FN3_NUM_X_ST_LOW_END         CHAR(1),
     2 PIWA2_FN3_LOW_B5SC(5)              CHAR(6),
     2 PIWA2_FN3_NUM_X_ST_HI_END          CHAR(1),
     2 PIWA2_FN3_HI_B5SC(5)               CHAR(6),
     2 PIWA2_FN3_REVERSAL_FLAG            CHAR(1),
     2 PIWA2_FN3_LIONKEY,
       3 PIWA2_FN3_LION_BORO              CHAR(1),
       3 PIWA2_FN3_LION_FACECODE          CHAR(4),
       3 PIWA2_FN3_LION_SEQ               CHAR(5),
     2 PIWA2_FN3_GENREC_FLAG              CHAR(1),
     2 PIWA2_FN3_SEG_LEN                  CHAR(5),
     2 PIWA2_FN3_SEG_SLOPE                CHAR(3),
     2 PIWA2_FN3_SEG_ORIENT               CHAR(1),
     2 PIWA2_FN3_MARBLE_RIKERS_FLAG       CHAR(1),
     2 FILLER_600                         CHAR(19),
     2 PIWA2_FN3_SEGMENT_ID               CHAR(7),
     2 PIWA2_FN3_DOT_SLA                  CHAR(1),
     2 PIWA2_FN3_CURVE_FLAG               CHAR(1),
     2 PIWA2_FN3_DOG_LEG                  CHAR(1),
     2 PIWA2_FN3_FEATURE_TYPE             CHAR(1),
     2 PIWA2_FN3_SEGMENT_TYPE             CHAR(1),
     2 FILLER_700                         CHAR(5),
     2 PIWA2_FN3_LEFT_SIDE_OF_STR,
       3 PIWA2_FN3_L_COM_DIST,
         4 PIWA2_FN3_L_COM_DIST_BORO      CHAR(1),
         4 PIWA2_FN3_L_COM_DIST_NUM       CHAR(2),
       3 PIWA2_FN3_L_LOW_HOUSENUM         CHAR(16),/*DISPLAY FORMAT*/
       3 PIWA2_FN3_L_HI_HOUSENUM          CHAR(16),/*DISPLAY FORMAT*/
       3 FILLER_800                       CHAR(33),/* FOR GSS USE*/
       3 PIWA2_FN3_L_ZIP                  CHAR(5),
       3 PIWA2_FN3_L_HEALTH_AREA          CHAR(4),
       3 PIWA2_FN3_L_POLICE_DIST,
         4  PIWA2_FN3_L_POL_PAT_BORO_CMD  CHAR(1),
         3  PIWA2_FN3_L_POL_PRECINCT      CHAR(3),
       3 PIWA2_FN3_L_FIRE_DIV             CHAR(2),
       3 PIWA2_FN3_L_FIRE_BAT             CHAR(2),
       3 PIWA2_FN3_L_FIRE_CO,
         4 PIWA2_FN3_L_FIRE_CO_TYPE       CHAR(1),
         4 PIWA2_FN3_L_FIRE_CO_NUM        CHAR(3),
       3 PIWA2_FN3_L_SCHL_DIST            CHAR(2),
       3 PIWA2_FN3_L_DYN_BLK              CHAR(3),
```

P2PL1 COPY File (continued)

```
      3 PIWA2_FN3_L_INSTRUCT_DIV         CHAR(2),
      3 FILLER_880                       CHAR(7),
      3 PIWA2_FN3_L_1990_CENS_TRCT       CHAR(6),
      3 PIWA2_FN3_L_2000_CENSUS_TRACT    CHAR(6),
      3 PIWA2_FN3_L_2000_CENSUS_BLOCK    CHAR(4),
      3 FILLER_890_RESV_DCP              CHAR(1),
      3 FILLER_900                       CHAR(30),
    2 PIWA2_FN3_RIGHT_SIDE_OF_STR,
      3 PIWA2_FN3_R_COM_DIST,
        4 PIWA2_FN3_R_COM_DIST_BORO      CHAR(1),
        4 PIWA2_FN3_R_COM_DIST_NUM       CHAR(2),
      3 PIWA2_FN3_R_LOW_HOUSENUM         CHAR(16),/*DISPLAY FORMAT*/
      3 PIWA2_FN3_R_HI_HOUSENUM          CHAR(16),/*DISPLAY FORMAT*/
      3 FILLER_1000                      CHAR(33),/*FOR GSS USE */
      3 PIWA2_FN3_R_ZIP                  CHAR(5),
      3 PIWA2_FN3_R_HEALTH_AREA          CHAR(4),
      3 PIWA2_FN3_R_POLICE_DIST,
        4  PIWA2_FN3_R_POL_PAT_BORO_CMD  CHAR(1),
        4  PIWA2_FN3_R_POL_PRECINCT      CHAR(3),
      3 PIWA2_FN3_R_FIRE_DIV             CHAR(2),
      3 PIWA2_FN3_R_FIRE_BAT             CHAR(2),
      3 PIWA2_FN3_R_FIRE_CO,
        4 PIWA2_FN3_R_FIRE_CO_TYPE       CHAR(1),
        4 PIWA2_FN3_R_FIRE_CO_NUM        CHAR(3),
      3 PIWA2_FN3_R_SCHL_DIST            CHAR(2),
      3 PIWA2_FN3_R_DYN_BLK              CHAR(3),
      3 PIWA2_FN3_R_INSTRUCT_DIV         CHAR(2),
      3 FILLER_1080                      CHAR(7),
      3 PIWA2_FN3_R_1990_CENS_TRCT       CHAR(6),
      3 PIWA2_FN3_R_2000_CENSUS_TRACT    CHAR(6),
      3 PIWA2_FN3_R_2000_CENSUS_BLOCK    CHAR(4),
      3 FILLER_890_RESV_DCP              CHAR(1),
      3 FILLER_1100                      CHAR(30);

DCL   PIWA2_FN3_L_COMDIST                CHAR(3)
        BASED(ADDR(PIWA2_FN3_L_COM_DIST));
DCL   PIWA2_FN3_L_POLDIST                CHAR(4)
        BASED(ADDR(PIWA2_FN3_L_POLICE_DIST));
DCL   PIWA2_FN3_R_COMDIST                CHAR(3)
        BASED(ADDR(PIWA2_FN3_R_COM_DIST));
DCL   PIWA2_FN3_R_POLDIST                CHAR(4)
        BASED(ADDR(PIWA2_FN3_R_POLICE_DIST));

 /*****************************************************************/
 /*******     FOR: FUNCTION 3C          *****************************/
DCL
  1 PIWA2_FUNCTION3C  BASED(PP2),
    2 PIWA2_FN3C_ACCESS_KEY              CHAR(21),
    2 PIWA2_FN3C_DUP_KEY_FLAG            CHAR(1),/*(OR CONT PARITY)*/
    2 PIWA2_FN3C_LOCATION_STATUS         CHAR(1),
    2 PIWA2_FN3C_COUNTY_BOUNDARY         CHAR(1),
    2 PIWA2_FN3C_PREF_LGC1               CHAR(2),
    2 PIWA2_FN3C_PREF_LGC2               CHAR(2),
    2 PIWA2_FN3C_PREF_LGC3               CHAR(2),
    2 PIWA2_FN3C_NUM_X_ST_LOW_END        CHAR(1),
    2 PIWA2_FN3C_LOW_B5SC(5)             CHAR(6),
    2 PIWA2_FN3C_NUM_X_ST_HI_END         CHAR(1),
    2 PIWA2_FN3C_HI_B5SC(5)              CHAR(6),
    2 PIWA2_FN3C_REVERSAL_FLAG           CHAR(1),
    2 PIWA2_FN3C_LIONKEY,
      3 PIWA2_FN3C_LION_BORO             CHAR(1),
      3 PIWA2_FN3C_LION_FACECODE         CHAR(4),
      3 PIWA2_FN3C_LION_SEQ              CHAR(5),
    2 PIWA2_FN3C_GENREC_FLAG             CHAR(1),
    2 PIWA2_FN3C_SEG_LEN                 CHAR(5),
    2 PIWA2_FN3C_SEG_SLOPE               CHAR(3),
    2 PIWA2_FN3C_SEG_ORIENT              CHAR(1),
    2 PIWA2_FN3C_MARBLE_RIKERS_FLAG      CHAR(1),
```

P2PL1 COPY File (continued)

```
     2 FILLER_1200                  CHAR(19),                     00027115
     2 PIWA2_FN3C_SEGMENT_ID        CHAR(7),                      00027215
     2 PIWA2_FN3C_DOT_SLA           CHAR(1),                      00027315
     2 PIWA2_FN3C_SIDE_OF_STR       CHAR(1),                      00027415
     2 PIWA2_FN3C_CURVE_FLAG        CHAR(1),                      00027515
     2 PIWA2_FN3C_FEATURE_TYPE      CHAR(1),                      00027615
     2 PIWA2_FN3C_SEGMENT_TYPE      CHAR(1),                      00027717
     2 FILLER_1300                  CHAR(5),                      00027817
     2 PIWA2_FN3C_BLOCKFACE_INFO,                                 00027917
       3 PIWA2_FN3C_COM_DIST,                                     00028017
         4 PIWA2_FN3C_COM_DIST_BORO    CHAR(1),                   00028117
         4 PIWA2_FN3C_COM_DIST_NUM     CHAR(2),                   00028217
       3 PIWA2_FN3C_LOW_HOUSENUM    CHAR(16),/*DISPLAY FORMAT*/   00028317
       3 PIWA2_FN3C_HI_HOUSENUM     CHAR(16),/*DISPLAY FORMAT*/   00028417
       3 PIWA2_FN3C_LOW_HOUSENUM2   CHAR(16),/*DISPLAY FORMAT*/   00028517
       3 PIWA2_FN3C_HI_HOUSENUM2    CHAR(16),/*DISPLAY FORMAT*/   00028617
       3 FILLER_1400                CHAR(1), /* FOR GSS USE */    00028717
       3 PIWA2_FN3C_ZIP             CHAR(5),                      00028817
       3 PIWA2_FN3C_HEALTH_AREA     CHAR(4),                      00028917
       3 PIWA2_FN3C_POLICE_DIST,                                  00029017
         4 PIWA2_FN3C_POL_PAT_BORO_CMD    CHAR(1),                00029117
         4 PIWA2_FN3C_POL_PRECINCT        CHAR(3),                00029217
       3 PIWA2_FN3C_FIRE_DIV        CHAR(2),                      00029317
       3 PIWA2_FN3C_FIRE_BAT        CHAR(2),                      00029417
       3 PIWA2_FN3C_FIRE_CO,                                      00029517
         4 PIWA2_FN3C_FIRE_CO_TYPE     CHAR(1),                   00029617
         4 PIWA2_FN3C_FIRE_CO_NUM      CHAR(3),                   00029717
       3 PIWA2_FN3C_SCHL_DIST       CHAR(2),                      00029817
       3 PIWA2_FN3C_DYN_BLK         CHAR(3),                      00029917
       3 PIWA2_FN3C_INSTRUCT_DIV    CHAR(2),                      00030017
       3 FILLER_1480                CHAR(7),                      00030117
       3 PIWA2_FN3C_1990_CENS_TRCT  CHAR(6),                      00030217
       3 PIWA2_FN3C_2000_CENSUS_TRACT   CHAR(6),                  00030317
       3 PIWA2_FN3C_2000_CENSUS_BLOCK   CHAR(4),                  00030417
       3 FILLER_1490_RESV_DCP       CHAR(01),                     00030517
       3 FILLER_1500                CHAR(30);                     00030617
                                                                  00030717
 DCL   PIWA2_FN3C_COMDIST           CHAR(3)                       00030817
         BASED(ADDR(PIWA2_FN3C_COM_DIST));                        00030917
 DCL   PIWA2_FN3C_POLDIST           CHAR(4)                       00031017
         BASED(ADDR(PIWA2_FN3C_POLICE_DIST));                     00031117
                                                                  00031217
                                                                  00031317
  /******************************************************************/ 00031417
  /*******     FOR: FUNCTION 5          *****************************/ 00031517
 DCL                                                              00031617
   1 PIWA2_FUNCTION5   BASED(PP2),                                00031717
     2 PIWA2_FN5_ADDR_MATCHING_KEY     CHAR(28),                  00031817
     2 FILLER_1600                     CHAR(172);                 00031917
                                                                  00032000
 PP2=ADDR(P2PL1);                                                 00040000
```

P2PL11A COPY File

```
/*******************************************************************/
/***  THIS IS THE PL/1 STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM  ***/
/***  INDEPENDENT REGULAR WORK AREA 2 FOR FUNCTIONS: 1A , BL,    ***/
/***  AND BN.                                                    ***/
/***  THESE THREE FUNCTIONS SHARE A SINGLE WORK AREA 2 LAYOUT.   ***/
/***                                                             ***/
/***  COPY FILE - P2PL11A.                          10/10/97     ***/
/*******************************************************************/

DCL
 1 P2PL11A,
   2 PIWA2_1A_ACCESS_KEY                CHAR(21),
   2 PIWA2_1A_CONT_PARITY               CHAR(1), /*(OR DUP ADDR IND)*/
   2 PIWA2_1A_LOW_HOUSENUM              CHAR(11), /* SORT FORMAT */
   2 PIWA2_1A_BBL,
     3 PIWA2_1A_BBL_BORO                CHAR(1),
     3 PIWA2_1A_BLOCK                   CHAR(5),
     3 PIWA2_1A_LOT                     CHAR(4),
   2 PIWA2_1A_LOT_VER                   CHAR(1),
   2 PIWA2_1A_SCC                       CHAR(1),
   2 FILLER_100                         CHAR(1),
   2 PIWA2_1A_GENERAL_LOT_INFO,
     3 PIWA2_1A_RPAD_BLDG_CLASS         CHAR(2),
     3 PIWA2_1A_CORNER_CODE             CHAR(2),
     3 PIWA2_1A_NUM_OF_STRUCTURES       CHAR(4),
     3 PIWA2_1A_NUM_OF_BLOCKFACES       CHAR(2),
     3 PIWA2_1A_INTERIOR_FLAG           CHAR(1),
     3 PIWA2_1A_VACANT_FLAG             CHAR(1),
     3 PIWA2_1A_IRREG_LOT_FLAG          CHAR(1),
   2 PIWA2_1A_MARBLE_RIKERS_FLAG        CHAR(1),
   2 PIWA2_1A_ADDR_LIST_OVFLOW_FLAG     CHAR(1),
   2 PIWA2_1A_STROLL_KEY,
     3 PIWA2_1A_STROLL_BORO             CHAR(1),
     3 PIWA2_1A_STROLL_5SC              CHAR(5),
     3 PIWA2_1A_STROLL_SIDE_OF_STR      CHAR(1),  /* L, R */
     3 PIWA2_1A_STROLL_HI_HOUSENUM      CHAR(11), /* SORT FORMAT */
     3 FILLER_200                       CHAR(1),
   2 FILLER_300                         CHAR(1),  /* FOR GSS USE*/
   2 PIWA2_1A_BIN                       CHAR(7),
   2 PIWA2_1A_CONDO_FLAG                CHAR(1),
   2 FILLER_400                         CHAR(1),
   2 PIWA2_1A_RPAD_CONDO_ID_NUM         CHAR(4),
   2 PIWA2_1A_CONDO_UNIT_ID_NUM         CHAR(7),
   2 PIWA2_1A_CONDO_BILL_BBL            CHAR(10),
   2 PIWA2_1A_CONDO_BILL_BBL_VER        CHAR(1),
   2 PIWA2_1A_CONDO_BILL_BBL_SCC        CHAR(1),
   2 PIWA2_1A_CONDO_LOW_BBL             CHAR(10),
   2 PIWA2_1A_CONDO_LOW_BBL_VER         CHAR(1),
   2 PIWA2_1A_CONDO_HIGH_BBL            CHAR(10),
   2 PIWA2_1A_CONDO_HIGH_BBL_VER        CHAR(1),
   2 FILLER_500                         CHAR(15),
   2 PIWA2_1A_COOP_NUM                  CHAR(4),
   2 PIWA2_1A_SANBORN,
     3 PIWA2_1A_SANBORN_BORO            CHAR(1),
     3 PIWA2_1A_SANBORN_VOL             CHAR(3),
     3 PIWA2_1A_SANBORN_PAGE            CHAR(4),
   2 PIWA2_1A_COMMERC_DIST              CHAR(5),
   2 PIWA2_1A_DOF_MAP_BORO              CHAR(1),
   2 PIWA2_1A_DOF_MAP_SECVOL            CHAR(4),
   2 PIWA2_1A_DOF_MAP_PAGE              CHAR(4),
   2 RESERVED_DCP                       CHAR(23),
   2 PIWA2_1A_X_COORD                   CHAR(07),
   2 PIWA2_1A_Y_COORD                   CHAR(07),
   2 FILLER_650                         CHAR(25),
   2 FILLER_700                         CHAR(8),  /* FOR GSS USE */
   2 PIWA2_1A_NUM_OF_ADDR               CHAR(4),
```

P2PL11A COPY File (continued)

```
    2 PIWA2_1A_ADDR_LIST(21),                                             00006704
      3 PIWA2_1A_LIST_LOW_HOUSENUM       CHAR(16), /*DISPLAY FORMAT*/    00006804
      3 PIWA2_1A_LIST_HI_HOUSENUM        CHAR(16), /*DISPLAY FORMAT*/    00006904
      3 PIWA2_1A_LIST_BORO               CHAR(1),                         00007004
      3 PIWA2_1A_LIST_5SC                CHAR(5),                         00007104
      3 PIWA2_1A_LIST_LGC                CHAR(2),                         00007204
      3 PIWA2_1A_LIST_BIN                CHAR(7),                         00007304
      3 PIWA2_1A_LIST_SIDE_OF_STR        CHAR(1),  /* L, R */            00007404
      3 PIWA2_1A_ADDR_TYPE               CHAR(1),  /* P=NAP, B=NAB  */   00007504
                                                   /* BLANK = NORMAL*/   00007604
      3 FILLER_800                       CHAR(4);                         00007704
                                                                          00007804
 DCL  PIWA2_1A_SANBORN_BVOLPAGE          CHAR(8)                          00007904
         BASED(ADDR(PIWA2_1A_SANBORN));                                   00008004
 DCL  PIWA2_1A_STROLLKEY                 CHAR(19)                         00008104
         BASED(ADDR(PIWA2_1A_STROLL_KEY));                                00009004
```

P2PL11AL COPY File

```
/*******************************************************************/
/***  THIS IS THE PL/1 STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM  ***/
/***  INDEPENDENT LONG WORK AREA 2 FOR FUNCTIONS: 1A , AND BL.   ***/
/***  THESE TWO FUNCTIONS SHARE A SINGLE LONG WORK AREA 2 LAYOUT.***/
/***                                                             ***/
/***  COPY FILE - P2PL11AL.                          10/11/2000  ***/
/*******************************************************************/

DCL
 1 P2PL11AL,
   2 PIWA2_1AL_ACCESS_KEY              CHAR(21),
   2 PIWA2_1AL_CONT_PARITY             CHAR(1), /*(OR DUP ADDR IND)*/
   2 PIWA2_1AL_LOW_HOUSENUM            CHAR(11), /* SORT FORMAT */
   2 PIWA2_1AL_BBL,
     3 PIWA2_1AL_BBL_BORO              CHAR(1),
     3 PIWA2_1AL_BLOCK                 CHAR(5),
     3 PIWA2_1AL_LOT                   CHAR(4),
   2 PIWA2_1AL_LOT_VER                 CHAR(1),
   2 PIWA2_1AL_SCC                     CHAR(1),
   2 FILLER_100                        CHAR(1),
   2 PIWA2_1AL_GENERAL_LOT_INFO,
     3 PIWA2_1AL_RPAD_BLDG_CLASS       CHAR(2),
     3 PIWA2_1AL_CORNER_CODE           CHAR(2),
     3 PIWA2_1AL_NUM_OF_STRUCTURES     CHAR(4),
     3 PIWA2_1AL_NUM_OF_BLOCKFACES     CHAR(2),
     3 PIWA2_1AL_INTERIOR_FLAG         CHAR(1),
     3 PIWA2_1AL_VACANT_FLAG           CHAR(1),
     3 PIWA2_1AL_IRREG_LOT_FLAG        CHAR(1),
   2 PIWA2_1AL_MARBLE_RIKERS_FLAG      CHAR(1),
   2 PIWA2_1AL_ADDR_LIST_OVFLOW_FLAG   CHAR(1),
   2 PIWA2_1AL_STROLL_KEY,
     3 PIWA2_1AL_STROLL_BORO           CHAR(1),
     3 PIWA2_1AL_STROLL_5SC            CHAR(5),
     3 PIWA2_1AL_STROLL_SIDE_OF_STR    CHAR(1),  /* L, R */
     3 PIWA2_1AL_STROLL_HI_HOUSENUM    CHAR(11), /* SORT FORMAT */
     3 FILLER_200                      CHAR(1),
   2 FILLER_300                        CHAR(1),  /* FOR GSS USE*/
   2 PIWA2_1AL_BIN                     CHAR(7),
   2 PIWA2_1AL_CONDO_FLAG              CHAR(1),
   2 FILLER_400                        CHAR(1),
   2 PIWA2_1AL_RPAD_CONDO_ID_NUM       CHAR(4),
   2 PIWA2_1AL_CONDO_UNIT_ID_NUM       CHAR(7),
   2 PIWA2_1AL_CONDO_BILL_BBL          CHAR(10),
   2 PIWA2_1AL_CONDO_BILL_BBL_VER      CHAR(1),
   2 PIWA2_1AL_CONDO_BILL_BBL_SCC      CHAR(1),
   2 PIWA2_1AL_CONDO_LOW_BBL           CHAR(10),
   2 PIWA2_1AL_CONDO_LOW_BBL_VER       CHAR(1),
   2 PIWA2_1AL_CONDO_HIGH_BBL          CHAR(10),
   2 PIWA2_1AL_CONDO_HIGH_BBL_VER      CHAR(1),
   2 FILLER_500                        CHAR(15),
   2 PIWA2_1AL_COOP_NUM                CHAR(4),
   2 PIWA2_1AL_SANBORN,
     3 PIWA2_1AL_SANBORN_BORO          CHAR(1),
     3 PIWA2_1AL_SANBORN_VOL           CHAR(3),
     3 PIWA2_1AL_SANBORN_PAGE          CHAR(4),
   2 PIWA2_1AL_COMMERC_DIST            CHAR(5),
   2 PIWA2_1AL_DOF_MAP_BORO            CHAR(1),
   2 PIWA2_1AL_DOF_MAP_SECVOL          CHAR(4),
   2 PIWA2_1AL_DOF_MAP_PAGE            CHAR(4),
   2 FILLER_600                        CHAR(23),
   2 PIWA2_1AL_X_COORD                 CHAR(07),
   2 PIWA2_1AL_Y_COORD                 CHAR(07),
   2 FILLER_650                        CHAR(25),
   2 FILLER_700                        CHAR(8),  /* FOR GSS USE*/
   2 PIWA2_1AL_NUM_OF_BINS             CHAR(4),
   2 PIWA2_1AL_BINS(2500)              CHAR(7);
```

P2PL11AL COPY File (continued)

```
                                                                 00006705
 DCL  PIWA2_1AL_SANBORN_BVOLPAGE         CHAR(8)                 00006805
         BASED(ADDR(PIWA2_1AL_SANBORN));                         00006905
 DCL  PIWA2_1AL_STROLLKEY                CHAR(19)                00007005
         BASED(ADDR(PIWA2_1AL_STROLL_KEY));                      00008005
```

P2PL13S COPY File

```
/*******************************************************************/
/***  THIS IS THE PL/1 STRUCTURE FOR GEOSUPPORT SYSTEM PLATFORM  ***/
/***  INDEPENDENT WORK AREA 2 FOR FUNCTION: 3S.                  ***/
/***                                                             ***/
/**   COPY FILE - P2PL13S.                             09/17/97  ***/
/*******************************************************************/
DCL
 1 P2PL13S,
   2 PIWA2_3S_ACCESS_KEY,
     3 FILLER_GSS                    CHAR(2),
     3 PIWA2_3S_PORS_STNAME_IND      CHAR(1),/* P = PRIMARY */
                                             /* S = SECONDARY */
     3 PIWA2_3S_BORO                 CHAR(1),
     3 PIWA2_3S_5SC                  CHAR(5),
     3 PIWA2_3S_LGC                  CHAR(2),/* BLANK IF P IN */
     3 FILLER                        CHAR(10),/* POSITION 3 */
   2 PIWA2_3S_NUM_OF_INTERSECTS      CHAR(3),
   2 PIWA2_3S_LIST_OF_INTERSECTS(350),
     3 PIWA2_3S_MARBLE_RIKERS_FLAG   CHAR(1),
     3 PIWA2_3S_DISTANCE             CHAR(5),
     3 PIWA2_3S_GAP_FLAG             CHAR(1),
     3 FILLER_100                    CHAR(7),
     3 PIWA2_3S_NUM_OF_STR           CHAR(1),
     3 PIWA2_3S_B7SC(5)              CHAR(8);
```

C COPY File (COW)

PAC COPY File

```
#ifndef GEOSUPPORT
#define GEOSUPPORT
#ifdef __cplusplus
extern "C" {
#endif
 /******************************************************************/
 /*                                                                */
 /*           GeoSupport System C-Language Header File             */
 /*              for Platform-Independent Work Areas               */
 /*                                                                */
 /*                Last Updated: 16 May 2006                       */
 /*                                                                */
 /******************************************************************/

 /******************************************************************/
 /*                                                                */
 /*      Group Items Used in Platform-Independent Work Area 1      */
 /*                                                                */
 /******************************************************************/
typedef struct { char boro;             /* Borough Code            */
                 char SC10[10];         /* 10 Digit Street Code    */
                 char Street_name[32];  /* Street Name             */
               } STREET;

typedef union { char bbl[10];           /* Borough-Block-Lot       */
                struct { char boro;     /* Borough                 */
                         char block[5]; /* Tax Block               */
                         char lot[4];   /* Tax Lot                 */
                       } cas;
              } BBL;

typedef struct {
                 char func_code[2];      /* Function Code           */
                 char hse_nbr_disp[16];  /* House nbr in Disp form  */
                 char hse_nbr_hns[11];   /* House nbr in Sort form  */
                 char lohse_nbr_disp[16];/* Lo House nbr in Disp form*/
                 char lohse_nbr_hns[11]; /* Lo House nbr in Sort form*/
                 STREET sti[3];          /* Street Information      */
                 BBL bbli;               /* Borough-Block-Lot       */
                 char filler01;          /* Filler-Tax Lot Version # */
                 char bld_id[7];         /* Building Id Number (BIN) */
                 char comp_direction;    /* Compass Direction       */
                 char comp_direction2;   /* Compass Direction-Fn 3S */
                 char filler02[7];       /* Future Use              */
                 char platform_ind;      /* Must be equal to 'C'    */
                 char filler03[101];     /* Future Use              */

                        /* Flags that influence processing */

                 char long_WA_flag;      /* Long Work Area 2 Flag   */
                                         /* Next 2 fields not impl  */
                 char hse_nbr_justify;   /* Hse Nbr Justification Flg*/
                 char hnl[2];            /* Hse Nbr Normalization len*/
                 char hse_nbr_over_flag; /* Reserved for GSS Use    */
                 char snl[2];            /* Street Name Norm Length */
                 char st_name_norm;      /* Street Name Normalization*/
                                         /*   Format Flag           */
                 char expanded_format;   /* Expanded Format Flag    */
                 char roadbedrequest;    /* Roadbed Request Switch  */
                 char res_01;            /* Reserved for Internal Use*/
                 char filler04[35];      /* Future Use              */
                } INWA1;

typedef struct {
                 char boro_name[9];      /* Boro Name of First Street*/
                 char hse_nbr_disp[16];  /* House nbr in Normalized */
```

PAC COPY File (continued)

```
                                 /* Display form               */
                char hse_nbr_hns[11];    /* House number in Sort Form*/
                STREET sto[3];           /* Street Information       */
                BBL bblo;                /* Boro(len=1), Block(len=5)*/
                                         /* and Lot (len=4)-Normalizd*/
                char filler05;           /* Filler-Tax Lot Version # */
                char lo_hse_nbr_disp[16]; /* low Hse nbr - display  */
                char lo_hse_nbr_hns[11]; /* low Hse nbr - sort form */
                char bin[7];             /* Building Id Number       */
                char attrbytes[3];       /* NAP Identification Number*/
                char filler07[132];      /* Future Use               */
                char nap_id_nbr[6];      /* NAP Id Nbr - Not Impl.   */
                char int_use1;           /* Internal Use Only        */
                char reason_code;        /* Reason Code              */
                char filler08;           /* Future Use               */
                char warn_code[2];       /* Warning Ret. Code-NotImpl*/
                char ret_code[2];        /* GeoSupport Return Code   */
                char msg[80];            /* GeoSupport Message       */
                char nbr_names[2];       /* Nbr of Sreet Names       */
                char B_7SC[10][8];       /* 10 Boro+7-digit st codes */
                char st_names[10][32];   /* Up to 10 Street Names    */
              } OUTWA1;

 /******************************************************************/
 /*                                                                */
 /*                Platform-Independent Work Area 1                */
 /*                                                                */
 /******************************************************************/

typedef struct { INWA1 input;
                OUTWA1 output;
              } C_WA1;

 /***************************************************************/
 /*                                                             */
 /*     Group Items Used in Platform-Independent Work Area 2's  */
 /*                                                             */
 /***************************************************************/

typedef struct {                         /* LION KEY                 */
                char lion_boro;          /* LION Borough Code        */
                                         /* Differs from GeoSupport  */
                                         /* Borough Codes            */
                char face[4];            /* Face Code                */
                char seq[5];             /* Sequence Number          */
              } LION;

typedef struct {
                char nbr_sts;            /* Number of streets */
                char B5SC[5][6];         /* Boro+5 Street Code*/
              } St_list;

typedef struct { char lo_hse_nbr[16];    /* Low House Nbr-Disply form*/
                char hi_hse_nbr[16];     /* Hi House Nbr-Display form*/
                char B5SC[6];            /* Boro & 5 digit Str Code  */
                char lgc[2];             /* DCP Preferred Street LGC */
                char bld_id[7];          /* BIN of address range     */
                char sos_ind;            /* Side of Street Indicator */
                char adr_type;           /* Address type - P=NAP,    */
                                         /*    B=NAB, Blank=Normal   */
                char filler01[4];        /* Future Use               */

              } ADDR_RANGE;

typedef struct { char sanborn_boro;      /* Sanborn Borough Code     */
                char sanborn_vol[3];     /* Sanborn Volume           */
                char sanborn_page[4];    /* Sanborn Page             */
              } SANBORN;
```

PAC COPY File (continued)

```
typedef struct { char com_dist[3];       /* Community District       */
                 char lo_hse_nbr[16];    /* Low House Nbr-Disply form*/
                 char hi_hse_nbr[16];    /* Hi House Nbr-Display form*/
                 char filler01[32];      /* Future Use               */
                 char iaei;              /* Interim Ass'tance Elig   */
                                         /* Indicator                */
                 char zip_code[5];       /* Zip code for Street seg. */
                 char health_area[4];    /* Health Area              */
                 char police_boro_com;   /* Police Patrl Boro Command*/
                 char police_pre[3];     /* Police Precinct          */
                 char fire_divisn[2];    /* Fire Division            */
                 char fire_bat[2];       /* Fire Battalion           */
                 char fire_co_type;      /* Fire Company Type        */
                 char fire_co_nbr[3];    /* Fire Company Number      */
                 char com_schl_dist[2];  /* Community School District*/
                 char dynam_blk[3];      /* Dynamic Block            */
                 char instruc_div[2];    /* Instructional Division   */
                 char filler02[7];       /* Future Use               */
                 char cen_tract_90[6];   /* 1990 Census Tract        */
                 char cen_tract_00[6];   /* 2000 Census Tract        */
                 char cen_blk_00[4];     /* 2000 Census Block        */
                 char filler03[1 ];      /* Possible Census Blk Suff */
                 char filler04[30];      /* Future Use               */
               } SEGSIDE;

typedef struct { char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                                         /* Alternative Boro flag    */
                 char len[5];            /* Len in ft from prev node */
                 char gap_flag;          /* Gap Flag                 */
                 char node_nbr[7];       /* Node Number of Intersect */
                 char nbr_streets;       /* Nbr streets intersecting */
                 char B7SC[5][8];        /* Lowest B7SC at Intersect */
                                         /* is first and 2nd Lowest  */
                                         /* B7SC is next. Remaining  */
                                         /* B7SC's in no particular  */
                                         /* order.                   */
               } CROSS_STRS;

 /******************************************************************/
 /*                                                                */
 /*          Platform-Independent Work Area 2 for Function 1       */
 /*                                                                */
 /******************************************************************/

typedef struct { char filler01[21];
                 char cont_parity_ind;   /* Continuous Parity Ind.   */
                                         /* or Duplicate Address Ind.*/
                 char lo_hse_nbr[11];    /* Lo House nbr in Sort form*/
                 char hi_hse_nbr[11];    /* Hi House Nbr in Sort form*/
                 char lgc[2];            /* DCP or BOE Preferred LGC */
                 St_list st[2];          /* 1=Low and 2=High         */
                                         /* Nbr of cross streets at  */
                                         /* low house nbr end of st  */
                                         /* B5SCs of lo end cross st */
                 LION key;               /* LION Key - 10 Characters */
                 char sagr_flag;         /* Special Address Generated*/
                                         /* Record flag              */
                 char sos_ind;           /* Side of Street Indicator */
                 char seg_len[5];        /* Segment Length in Feet   */
                 char coord[3][7];       /* 1 = X coordinate,        */
                                         /* 2 = Y coordinate,        */
                                         /* 3 = Z coordinate, Not Imp*/
                 char iaei;              /* Interim Ass'tance Elig   */
                                         /* Indicator                */
                 char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                                         /* Alternative Borough flag */
                 char DOT_slca;          /* DOT St Lght Contractr Are*/
```

PAC COPY File (continued)

```
                                                                  02030017
            char com_dist[3];       /* Community District       */ 02040017
                                    /* Position 0 contains the  */ 02050017
                                    /* CD Boro Code & Pos 1 & 2,*/ 02060017
                                    /* the district number      */ 02070017
            char zip_code[5];       /* Zip code for st seg      */ 02080017
          /* Following seven fields   */                           02090017
          /* used for Function 1E only*/                           02100017
            char ed[3];             /* Election District        */ 02110017
            char ad[2];             /* Assembly District        */ 02120017
            char sped_flag;         /* Split Elect District Flag*/ 02130017
            char congress_dist[2];  /* Congressional District   */ 02140017
            char state_sen_dist[2]; /* State Senatorial District*/ 02150017
            char civil_court[2];    /* Civil Court District     */ 02160017
            char city_council[2];   /* City Council District    */ 02170017
            char health_cent[2];    /* Health Center District   */ 02180017
            char health_area[4];    /* Health Area              */ 02190017
            char sanit_dist[3];     /* Sanitation District      */ 02200017
            char sanit_sub_sect[2]; /* Sanit Collect Scheduling */ 02210017
                                    /* Section and Subsection   */ 02220017
            char sanit_reg_pick_up[5]; /* Regular Pick up       */ 02230017
            char sanit_recycle[3];  /* Recycle pick up          */ 02240017
            char police_boro_com;   /* Police Patrol Boro Commnd*/ 02250017
            char police_pre[3];     /* Police Precinct          */ 02260017
            char fire_divisn[2];    /* Fire Division            */ 02270017
            char fire_bat[2];       /* Fire Battalion           */ 02280017
            char fire_co_type;      /* Fire Company Type        */ 02290017
            char fire_co_nbr[3];    /* Fire Company Number      */ 02300017
            char scsd_flag;         /* Split Com School District*/ 02310017
                                    /* flag                     */ 02320017
            char com_schl_dist[2];  /* Community School District*/ 02330017
            char dynam_blk[3];      /* Dynamic Block            */ 02340017
            char instruc_div[2];    /* Instructional Division   */ 02350017
            char feature_type;      /* Feature Type Code        */ 02360019
            char segmenttypecode;   /* Segment Type Code        */ 02370022
            char alx;               /* Segment split by Alley(s)*/ 02371024
                                    /* A=Split by Alley(s)      */ 02372024
                                    /* X=Cross Streets Modified */ 02373024
            char filler02[4];       /* Future Use               */ 02380025
            char cen_tract_90[6];   /* 1990 Census Tract        */ 02390017
            char cen_tract_00[6];   /* 2000 Census Tract        */ 02400017
            char cen_blk_00[4];     /* 2000 Census Block        */ 02410017
            char filler04[40];      /* Future Use               */ 02420021
            char true_hns[11];      /* Underlying HNS           */ 02430021
            char true_b7sc[8];      /* True Boro 7 Street Code  */ 02440021
            char seg_id[7];         /* Segment Identifier       */ 02450021
            char curv_flag;         /* Curve Flag               */ 02460021
          } C_WA2_F1;                                              02470021
                                                                   02480021
 /******************************************************************/ 02490021
 /*                                                                */ 02500021
 /*      Platform-Independent Work Area 2 for Function 1A          */ 02510021
 /*                                                                */ 02520021
 /******************************************************************/ 02530021
                                                                   02540021
typedef struct { char filler01[21];                                02550021
            char cont_parity_ind;   /* Continuous Parity Ind    */ 02560021
                                    /* or Duplicate Address Ind */ 02570021
            char lo_hse_nbr[11];    /* Low House Number-Sort Frm*/ 02580021
                                                                   02590021
            BBL bbl;                /* Borough-Block-Lot        */ 02600021
            char filler02;          /* Reserved for Tax Lot Ver#*/ 02610021
            char RPAD_scc;          /* RPAD Self_Check Code(SCC)*/ 02620021
            char filler03;                                         02630021
            char RPAD_lucc[2];      /* RPAD Land Use Class. Code*/ 02640021
            char corner[2];         /* Corner Code              */ 02650021
            char nbr_blds[4];       /* Nbr of buildings on lot  */ 02660021
            char nbr_str[2];        /* Nbr Street Frontages     */ 02670021
```

PAC COPY File (continued)

```
                char inter_flag;        /* Interior Lot Flag        */
                char vacant_flag;       /* Vacant Lot Flag          */
                char irreg_flag;        /* Irregularly-Shaped Lot Fl*/
                char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                char adr_range_overflow;/* Addr Rnge Lst Ovrflow Flg*/
                char stroll_key[18];    /* Strolling key            */
                char filler04;
                char res_internal_use;  /* Reserved for Internal Use*/
                char bld_id[7];         /* Building Ident. Number   */
                                        /* (BIN) of Input Address of*/
                                        /* Existing Building, If any*/
                char condo_flag;        /* Condominium Flag         */
                char filler05;          /* Future Use               */
                char condo_id[4];       /* RPAD Condo Id Number     */
                char condo_unit_id[7];  /* Condo Unit Id Nbr-Not Impl*/
                BBL  condo_bill_bbl;    /* Condo Billing BBL        */
                char filler06;          /* Reserved for Tax Lot Ver */
                char condo_scc;         /* Self-Check Code          */
                BBL  condo_lo_bbl;      /* Low BBL of Condo         */
                char filler07;          /* Reserved for Tax Lot Ver */
                BBL  condo_hi_bbl;      /* High BBL of Condo        */
                char filler08;          /* Reserved for Tax Lot Ver */
                char filler09[15];
                char co_op_nbr[4];      /* Co-op Number             */
                SANBORN San;            /* Sanborn Information      */
                char business_area[5];  /* Business Area            */
                char tax_map_nbr[5];    /* Tax Map Nbr-Sect and Vol */
                char filler10[4 ];      /* Tax Map Nbr Page Not Impl*/
                char filler11[23];
                char coord[2][7];       /* 1 = X coordinate-Annotat */
                                        /* 2 = Y coordinate-Annotat */
                char filler12[25];
                char int_use[8];        /* Internal Use             */
                char nbr_addr[4];       /* Nbr of Addr Ranges or Nbr*/
                                        /* of BINs in List          */
                union {
                        ADDR_RANGE addr_range[21]; /* List of Addr  */
                        char bin_list[2500][7];    /* Ranges or BINs*/
                      } bar;
              } C_WA2_F1A;

 /*****************************************************************/
 /*                                                               */
 /*       Platform-Independent Work Area 2 for Function 2         */
 /*                                                               */
 /*****************************************************************/

typedef struct { char filler01[21];
                char rep_cnt;           /* Intersection Replication */
                                        /* Counter*/
                char lgc[2][2];         /* Preferred LGCs           */
                St_list inter;          /* Number of Intersecting St*/
                                        /* B5SCs of Intersection St */
                char Dup_comp;          /* Duplicate compass Directn*/
                char filler02[5 ];
                char LION_node_nbr[7 ]; /* LION Node Number         */
                char coord[3][7];       /* 1 = X coordinate,        */
                                        /* 2 = Y coordinate,        */
                                        /* 3 = Z coordinate, Not Imp*/
                SANBORN San[2];         /* Sanborn Information      */
                char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                char DOT_slca;          /* DOT St Lght Contractr Are*/
                char com_dist[3];       /* Community District       */
                char zip_code[5];       /* Zip code for st segment  */
                char health_area[4];    /* Health Area              */
                char police_boro_com;   /* Police Patrol Boro Commnd*/
                char police_pre[3];     /* Police Precinct          */
                char fire_sector[2];    /* Fire Sector              */
```

PAC COPY File (continued)

```
                 char fire_bat[2];       /* Fire Battalion           */
                 char fire_co_type;      /* Fire Company Type        */
                 char fire_co_nbr[3];    /* Fire Company Number      */
                 char com_schl_dist[2];  /* Community School District*/
                 char cen_tract_00[6];   /* 2000 Census Tract        */
                 char cen_tract_90[6];   /* 1990 Census Tract        */
                 char level_codes [10];  /* Level codes              */
                 char instruc_div [2];   /* Instructional Division   */
                 char filler03[41];
               } C_WA2_F2;

 /*****************************************************************/
 /*                                                               */
 /*        Platform-Independent Work Area 2 for Function 3        */
 /*                                                               */
 /*****************************************************************/

typedef struct { char filler01[21];
                 char dup_key_flag;      /* Duplicate Key Flag or    */
                                         /* Continuous Parity Flag   */
                 char loc_stat_seg;      /* Locational Status of Seg */
                 char cnty_bnd_ind;      /* County Boundary Indicat  */
                 char lgc[3][2];         /* Preferred LGCs           */
                 St_list st[2];          /* 1=Low and 2=High         */
                                         /* Nbr of cross sts at low  */
                                         /* house nbr end of street  */
                                         /* B5SCs of lo end X sts    */
                 char x_street_reversal_flag; /* X St Reversal Flag  */
                 LION key;               /* LION Key                 */
                 char genr_flag;         /* Generated Record Flag    */
                 char seg_len[5];        /* Segment Length in Feet   */
                 char seg_azm[3];        /* Segment Azimuth          */
                 char seg_orient;        /* Segment Orientation      */
                 char mh_ri_flag;        /* Marble Hill/Rikers Island*/
                                         /* Alternative Boro flag    */
                 char filler02[19];      /* Future use               */
                 char seg_id[7];         /* Segment Identifier       */
                 char DOT_slca;          /* DOT St Lght Contractr Are*/
                 char curve_flag;        /* Curve Flag               */
                 char dog_leg;           /* Dog leg flag             */
                 char feature_type;      /* Feature Type Code        */
                 char segmenttypecode;   /* Segment Type Code        */
                 char filler03[5];
                 SEGSIDE side[2];        /* 1 = Left Side of street  */
                                         /* 2 = Right Side of street */
               } C_WA2_F3;

 /*****************************************************************/
 /*                                                               */
 /*        Platform-Independent Work Area 2 for Function 3C       */
 /*                                                               */
 /*****************************************************************/

typedef struct { char filler01[21];
                 char dup_key_flag;      /* Duplicate Key Flag or    */
                                         /* Continuous Parity Flag   */
                 char loc_stat_seg;      /* Locational Status of Seg */
                 char cnty_bnd_ind;      /* County Boundary Indicat  */
                 char lgc[3][2];         /* Preferred LGCs           */
                 St_list st[2];          /* 1=Low and 2=High         */
                                         /* Nbr of cross sts at low  */
                                         /* house nbr end of street  */
                                         /* B5SCs of lo end Cross sts*/
                 char x_street_reversal_flag; /* X St Reversal Flag  */
                 LION key;               /* LION key                 */
                 char genr_flag;         /* Generated Record Flag    */
                 char seg_len[5];        /* Segment Length in Feet   */
                 char seg_azm[3];        /* Segment Azimuth          */
```

PAC COPY File (continued)

```
                 char seg_orient;         /* Segment Orientation      */
                 char mh_ri_flag;         /* Marble Hill/Rikers Island*/
                                          /* Alternative Boro flag    */
                 char filler02[19];       /* Future use               */
                 char seg_id  [7];        /* Segment Identifier       */
                 char DOT_slca;           /* DOT St Lght Contractr Are*/
                 char sos_ind;            /* Side of Street Indicator */
                 char curve_flag;         /* Curve Flag               */
                 char feature_type;       /* Feature Type Code        */
                 char segmenttypecode;    /* Segment Type Code        */
                 char filler03[5];
                 SEGSIDE req;            /* Geographic Information for*/
                                          /* Requested Side of segment*/
              } C_WA2_F3C;

 /*******************************************************************/
 /*                                                                 */
 /*        Platform-Independent Work Area 2 for Function 3S        */
 /*                                                                 */
 /*******************************************************************/

typedef struct { char filler01[21];
                 char nbr_x_str[3];       /* Nbr of Cross sts in list */
                 CROSS_STRS cross_strs[350];/* Cross Street structure*/
               } C_WA2_F3S;

#ifdef __cplusplus
       }
#endif
#endif
```

NATURAL LDAs (COW)

GEOLP1 COPY File

```
*     USER PROGRAMS MUST RESET GEOLP1 BEFORE PRIMING WORKAREA 1
   1 GEOLP1                                      /* LRECL=1200
*     THE FIELD P1NAT IS USED AS A PARAMETER TO CALL GEOSUPPORT
   2 P1NAT                          A           2
R  2 P1NAT
*  * * * *   INPUT FIELDS    * * * *  *
   3 PIWA1-IN-FUNCTION-CODE         A           2 /* BEGINNING OF FCT 1 LAYOUT
R  3 PIWA1-IN-FUNCTION-CODE
   4 PIWA1-IN-FUNCTION-1            A           1
   4 PIWA1-IN-FUNCTION-2            A           1
   2 PIWA1-IN-HOUSENUM-DISPLAY      A          16
   2 PIWA1-IN-HOUSENUM-SORT         A          11
   2 PIWA1-IN-LOW-HOUSENUM-DISPLAY  A          16
   2 PIWA1-IN-LOW-HOUSENUM-SORT     A          11
   2 PIWA1-IN-BORO-1                A           1
   2 PIWA1-IN-10SC-1                A          10
   2 PIWA1-IN-STREET-1              A          32
   2 PIWA1-IN-BORO-2                A           1
   2 PIWA1-IN-10SC-2                A          10
   2 PIWA1-IN-STREET-2              A          32
   2 PIWA1-IN-BORO-3                A           1
   2 PIWA1-IN-10SC-3                A          10
   2 PIWA1-IN-STREET-3              A          32
   2 PIWA1-IN-BBL                   A          10
R  2 PIWA1-IN-BBL
   3 PIWA1-IN-BBL-BORO              A           1
   3 PIWA1-IN-BLOCK                 A           5
   3 PIWA1-IN-LOT                   A           4
   2 FILLER-50                      A           1 /* FUTURE LOT VERSION #
   2 PIWA1-IN-BIN                   A           7
   2 PIWA1-IN-COMPASS               A           1
   2 PIWA1-IN-COMPASS2              A           1
   2 FILLER-100                     A           7
   2 PIWA1-IN-PLATFORM-INDICATOR    A           1 /* C= C LANG
   2 FILLER-200                     A         101
   2 PIWA1-IN-LONG-WORKAREA2-FLAG   A           1 /* L=LONG WA - 1A/BL(1200)
   2 PIWA1-IN-HSE-NBR-JUSTIFY       A           1
   2 PIWA1-IN-HNL                   A           2 /* HN LENGTH
   2 PIWA1-IN-HSE-OVER-FLAG         A           1 /* HN OVERRIDE *,$,' '.
   2 PIWA1-IN-SNL                   A           2
   2 PIWA1-IN-SN-NORM-FORMAT        A           1 /* C=COMPACT, S OR' '=SORTT
   2 PIWA1-IN-EXPANDED-FORMAT       A           1
   2 PIWA1-IN-ROADBED-REQ-SWITCH    A           1
   2 PIWA1-IN-INTERNAL-USE-LEGACY   A           1 /* RESERVED FOR GSS USE
   2 FILLER-400                     A          35
*  * * * *   OUTPUT FIELDS   * * * *  *
   2 PIWA1-OUT-BORONAME             A           9
   2 PIWA1-OUT-HOUSENUM-DISPLAY     A          16
   2 PIWA1-OUT-HOUSENUM-SORT        A          11
   2 PIWA1-OUT-B10SC-1              A          11
   2 PIWA1-OUT-STREET-1             A          32
   2 PIWA1-OUT-B10SC-2              A          11
   2 PIWA1-OUT-STREET-2             A          32
   2 PIWA1-OUT-B10SC-3              A          11
   2 PIWA1-OUT-STREET-3             A          32
   2 PIWA1-OUT-BBL                  A          10
R  2 PIWA1-OUT-BBL
   3 PIWA1-OUT-BBL-BORO             A           1
   3 PIWA1-OUT-BLOCK                A           5
   3 PIWA1-OUT-LOT                  A           4
   2 FILLER-LOT-VERSION             A           1 /* FOR FUTRUE LOT VERSION #
   2 PIWA1-OUT-LOW-HOUSENUM-DISPLAY A          16
   2 PIWA1-OUT-LOW-SORT             A          11
   2 PIWA1-OUT-BIN                  A           7
   2 PIWA1-OUT-STREET-ATTR          A           1 (1:3)
   2 FILLER-500                     A         138
```

GEOLP1 COPY File (continued)

```
2 PIWA1-OUT-SND-ATTR               A           1
2 PIWA1-OUT-REASON-CODE            A           1
2 FILLER-600                       A           1
2 PIWA1-OUT-WARNING-CODE           A           2
2 PIWA1-OUT-RETURN-CODE            A           2
2 PIWA1-OUT-ERROR-MESSAGE          A          80
2 PIWA1-OUT-NUM-SIMILAR-STRS       A           2
2 PIWA1-OUT-SIMILAR-B7SC           A           8 (1:10)
2 PIWA1-OUT-SIMILAR-NAMES          A          32 (1:10)
```

GEOLP2 COPY File

```
    1 GEOLP2
*     THE FIELD P2NAT IS USED AS A PARAMETER TO CALL GEOSUPPORT FOR ALL
*         FUNCTIONS THAT ARE REDEFINED ON GEOLP2
   2 P2NAT                            A         21
R  2 P2NAT
*  *  BEGINNING OF FUNCTION 1 LAYOUT  *       **** *******
   3 PIWA2-FN1-ACCESS-KEY             A         21
   2 PIWA2-FN1-CONT-PARITY            A          1 /* (OR DUP ADDR IND)
   2 PIWA2-FN1-LOW-HOUSENUM           A         11 /* SORT FORMAT
   2 PIWA2-FN1-HI-HOUSENUM            A         11 /* SORT FORMAT
   2 PIWA2-FN1-PREFERRED_LGC          A          2
   2 PIWA2-FN1-NUM-X-ST-LOW-END       A          1
   2 PIWA2-FN1-LOW-B5SC               A          6 (1:5) /* 30-BYTES
   2 PIWA2-FN1-NUM-X-ST-HI-END        A          1
   2 PIWA2-FN1-HI-B5SC                A          6 (1:5) /* 30-BYTES
   2 PIWA2-FN1-LIONKEY                A         10
R  2 PIWA2-FN1-LIONKEY
   3 PIWA2-FN1-LION-BORO              A          1
   3 PIWA2-FN1-LION-FACECODE          A          4
   3 PIWA2-FN1-LION-SEQ               A          5
   2 PIWA2-FN1-SPECIAL-ADDR-FLAG      A          1
   2 PIWA2-FN1-SIDE-OF-STR            A          1
   2 PIWA2-FN1-SEG-LEN                A          5
   2 PIWA2-FN1-XCOORD                 A          7
   2 PIWA2-FN1-YCOORD                 A          7
   2 FILLER-100                       A          7 /* FOR ZCOORD
   2 FILLER-200                       A          1 /* FOR GSS USE
   2 PIWA2-FN1-MARBLE-RICKERS-FLAG    A          1
   2 PIWA2-FN1-DOT-SLA                A          1
   2 PIWA2-FN1-COM-DIST               A          3
R  2 PIWA2-FN1-COM-DIST
   3 PIWA2-FN1-COM-DIST-BORO          A          1
   3 PIWA2-FN1-COM-DIST-NUM           A          2
   2 PIWA2-FN1-ZIP                    A          5
*  *                                  *       **** *****
*  *  THE FN1E FIELDS ARE VALID ONLY  *       **** *****
*  *  FOR FUNCTION 1E, NOT FUNC 1.    *       **** *****
   2 PIWA2-FN1E-ELECT-DIST            A          3
   2 PIWA2-FN1E-ASSEM-DIST            A          2
   2 PIWA2-FN1E-SPLIT-ED-FLAG         A          1
   2 PIWA2-FN1E-CONG-DIST             A          2
   2 PIWA2-FN1E-SENATE-DIST           A          2
   2 PIWA2-FN1E-COURT-DIST            A          2
   2 PIWA2-FN1E-COUNCIL-DIST          A          2
*  *                                  *       **** *****
   2 PIWA2-FN1-HEALTH-CENTER-DIST     A          2
   2 PIWA2-FN1-HEALTH-AREA            A          4
   2 PIWA2-FN1-SANI-DIST              A          3
R  2 PIWA2-FN1-SANI-DIST
   3 PIWA2-FN1-SANI-DIST-BORO         A          1
   3 PIWA2-FN1-SANI-DIST-NUM          A          2
   2 PIWA2-FN1-SANI-SUBSEC            A          2
   2 PIWA2-FN1-SANI-REG               A          5
   2 PIWA2-FN1-SANI-REC               A          3
   2 PIWA2-FN1-POLICE-DIST            A          4
R  2 PIWA2-FN1-POLICE-DIST
   3 PIWA2-FN1-POL-PAT-BORO-CMD       A          1
   3 PIWA2-FN1-POL-PRECINCT           A          3
   2 PIWA2-FN1-FIRE-DIV               A          2
   2 PIWA2-FN1-FIRE-BAT               A          2
   2 PIWA2-FN1-FIRE-CO                A          4
R  2 PIWA2-FN1-FIRE-CO
   3 PIWA2-FN1-FIRE-CO-TYPE           A          1
   3 PIWA2-FN1-FIRE-CO-NUM            A          3
   2 PIWA2-FN1-SCHL-DIST-SPLIT-FLAG   A          1
   2 PIWA2-FN1-SCHL-DIST              A          2
```

GEOLP2 COPY File (continued)

```
   2 PIWA2-FN1-DYN-BLK                A           3
   2 PIWA2-FN1-INSTRUCT-DIV           A           2
   2 PIWA2-FN1-FEATURE-TYPE           A           1
   2 PIWA2-FN1-SEGMENT-TYPE           A           1
   2 PIWA2-FN1-ALX                    A           1
   2 FILLER-290                       A           4
   2 PIWA2-FN1-1990-CENS-TRCT         A           6
   2 PIWA2-FN1-2000-CENS-TRCT         A           6
   2 PIWA2-FN1-2000-CNES-BLK          A           4
   2 FILLER-300                       A          40
   2 PIWA2-FN1-REAL-HNS               A          11 /* UNDERLYING HNS
   2 PIWA2-FN1-REAL-B7SC              A           8
   2 PIWA2-FN1-SEGMENT-ID             A           7
   2 PIWA2-FN1-CURVE-FLAG             A           1
*  *  END OF FUNCTION 1 LAYOUT        *        **** *******
*  - ------------------------------- -        ---- --------------------------------
*  *  BEGINNING OF FUNCTION 3C LAYOUT          **** ********
R  1 GEOLP2
   2 PIWA2-FN3C-ACCESS-KEY            A          21
   2 PIWA2-FN3C-DUP-KEY-FLAG          A           1 /* OR FN3C CONTI PARITY
   2 PIWA2-FN3C-LOCATION-STATUS       A           1
   2 PIWA2-FN3C-COUNTY-BOUNDARY       A           1
   2 PIWA2-FN3C-PREFERRED-LGC1        A           2
   2 PIWA2-FN3C-PREFERRED-LGC2        A           2
   2 PIWA2-FN3C-PREFERRED-LGC3        A           2
   2 PIWA2-FN3C-NUM-X-ST-LOW-END      A           1
   2 PIWA2-FN3C-LOW-B5SC              A           6 (1:5) /* 30-BYTES
   2 PIWA2-FN3C-NUM-X-ST-HI-END       A           1
   2 PIWA2-FN3C-HI-B5SC               A           6 (1:5) /* 30-BYTES
   2 PIWA2-FN3C-REVERSAL-FLAG         A           1
   2 PIWA2-FN3C-LIONKEY               A          10
R  2 PIWA2-FN3C-LIONKEY
   3 PIWA2-FN3C-LION-BORO             A           1
   3 PIWA2-FN3C-LION-FACECODE         A           4
   3 PIWA2-FN3C-LION-SEQ              A           5
   2 PIWA2-FN3C-GENREC-FLAG           A           1
   2 PIWA2-FN3C-SEG-LEN               A           5
   2 PIWA2-FN3C-SEG-SLOPE             A           3
   2 PIWA2-FN3C-SEG-ORIENT            A           1
   2 PIWA2-FN3C-MARBLE-RIKERS-FLAG    A           1
   2 FILLER-1200                      A          19
   2 PIWA2-FN3C-SEGMENT-ID            A           7
   2 PIWA2-FN3C-DOT-SLA               A           1
   2 PIWA2-FN3C-SIDE-OF-STR           A           1
   2 PIWA2-FN3C-CURVE-FLAG            A           1
   2 PIWA2-FN3C-FEATURE-TYPE          A           1
   2 PIWA2-FN3C-SEGMENT-TYPE          A           1
   2 FILLER-1300                      A           5
*  * PIWA2-FN3C-BLOCKFACE-INFO                 **** *****
   2 PIWA2-FN3C-COM-DIST              A           3
R  2 PIWA2-FN3C-COM-DIST
   3 PIWA2-FN3C-COMDIST-BORO          A           1
   3 PIWA2-FN3C-COMDIST-NUM           A           2
   2 PIWA2-FN3C-LOW-HOUSENUM          A          16 /* DISPLAY FORMAT
   2 PIWA2-FN3C-HI-HOUSENUM           A          16 /* DISPLAY FORMAT
   2 PIWA2-FN3C-LOW-HOUSENUM2         A          16 /* DISPLAY FORMAT
   2 PIWA2-FN3C-HI-HOUSENUM2          A          16 /* DISPLAY FORMAT
   2 FILLER-1400                      A           1 /* FOR GSS USE
   2 PIWA2-FN3C-ZIP                   A           5
   2 PIWA2-FN3C-HEALTH-AREA           A           4
   2 PIWA2-FN3C-POLICE-DIST           A           4
R  2 PIWA2-FN3C-POLICE-DIST
   3 PIWA2-FN3C-POL-PAT-BORO-CMD      A           1
   3 PIWA2-FN3C-POL-PRECINCT          A           3
   2 PIWA2-FN3C-FIRE-DIV              A           2
   2 PIWA2-FN3C-FIRE-BAT              A           2
   2 PIWA2-FN3C-FIRE-CO               A           4
R  2 PIWA2-FN3C-FIRE-CO
```

GEOLP2 COPY File (continued)

```
   3 PIWA2-FN3C-FIRE-CO-TYPE          A          1
   3 PIWA2-FN3C-FIRE-CO-NUM           A          3
   2 PIWA2-FN3C-SCHL-DIST             A          2
   2 PIWA2-FN3C-DYN-BLK               A          3
   2 PIWA2-FN3C-INSTRUCT-DIV          A          2
   2 FILLER-1480                      A          7
   2 PIWA2-FN3C-1900-CENS-TRCT        A          6
   2 PIWA2-FN3C-2000-CENS-TRCT        A          6
   2 PIWA2-FN3C-2000-CENS-BLK         A          4
   2 FILLER-1490                      A          1 /* POSSIBLE CENSUS BLK SFX
   2 FILLER-1500                      A         30
*  *  END OF FUNCTION 3C LAYOUT       *       **** *******
*  - -------------------------------- -       ---- --------------------------------
*  *  BEGINNING OF FUNCTION 5 LAYOUT  *       **** *******
R  1 GEOLP2
   2 PIWA2-FN5-ADDR-MATCHING-KEY      A         28
*  *  END OF FUNCTION 5 LAYOUT        *       **** *******
*  - -------------------------------- -       ---- --------------------------------
```

GEOLP22 COPY File

```
     1 GEOLP22
*    * THE FIELD P2NAT IS USED AS A PARAMETER TO CALL GEOSUPPORT
     2 P2NAT2                             A          21
R    2 P2NAT2
     3 PIWA2-FN2-ACCESS-KEY               A          21
     2 PIWA2-FN2-DUP-INTERSECT-FLAG       A           1
     2 PIWA2-FN2-PREFERRED-LGC1           A           2
     2 PIWA2-FN2-PREFERRED-LGC2           A           2
     2 PIWA2-FN2-NUM-OF-INTERSECTS        A           1
     2 PIWA2-FN2-INTERSECT-B5SC           A           6 (1:5) /* 30-BYTES
     2 PIWA2-FN2-COMPDIR                  A           1
     2 FILLER-350                         A           5
     2 PIWA2-FN2-LION-NODE-NUM            A           7
     2 PIWA2-FN2-XCOORD                   A           7
     2 PIWA2-FN2-YCOORD                   A           7
     2 FILLER-400                         A           7 /* FOR ZCOORD
     2 PIWA2-FN2-SANBORN1                 A           8
R    2 PIWA2-FN2-SANBORN1
     3 PIWA2-FN2-SANBORN1-BORO            A           1
     3 PIWA2-FN2-SANBORN1-VOL             A           3
     3 PIWA2-FN2-SANBORN1-PAGE            A           4
     2 PIWA2-FN2-SANBORN2                 A           8
R    2 PIWA2-FN2-SANBORN2
     3 PIWA2-FN2-SANBORN2-BORO            A           1
     3 PIWA2-FN2-SANBORN2-VOL             A           3
     3 PIWA2-FN2-SANBORN2-PAGE            A           4
     2 PIWA2-FN2-MARBLE-RIKERS-FLAG       A           1
     2 PIWA2-FN2-DOT-SLA                  A           1
     2 PIWA2-FN2-COM-DIST                 A           3
R    2 PIWA2-FN2-COM-DIST
     3 PIWA2-FN2-COM-DIST-BORO            A           1
     3 PIWA2-FN2-COM-DIST-NUM             A           2
     2 PIWA2-FN2-ZIP                      A           5
     2 PIWA2-FN2-HEALTH-AREA              A           4
     2 PIWA2-FN2-POLICE-DIST              A           4
R    2 PIWA2-FN2-POLICE-DIST
     3 PIWA2-FN2-POL-PAT-BORO-CMD         A           1
     3 PIWA2-FN2-POL-PRECINCT             A           3
     2 PIWA2-FN2-FIRE-DIV                 A           2
     2 PIWA2-FN2-FIRE-BAT                 A           2
     2 PIWA2-FN2-FIRE-CO                  A           4
R    2 PIWA2-FN2-FIRE-CO
     3 PIWA2-FN2-FIRE-CO-TYPE             A           1
     3 PIWA2-FN2-FIRE-CO-NUM              A           3
     2 PIWA2-FN2-SCHL-DIST                A           2
     2 PIWA2-FN2-2000-CENS-TRCT           A           6
     2 PIWA2-FN2-1900-CENS-TRCT           A           6
     2 PIWA2-FN2-LEVEL-CODE-TBL           A          10
R    2 PIWA2-FN2-LEVEL-CODE-TBL
     3 PIWA2-FN2-LEVEL-CODE               A           1 (5,2) /* 10-BYTES
     2 PIWA2-FN2-INSTRUCT-DIV             A           2
     2 FILLER-500                         A          41
```

GEOLP23 COPY File

```
   1 GEOLP23
*    THE FIELD P2NAT3 IS USED AS A PARAMETER TO CALL GEOSUPPORT
   2 P2NAT3                          A          21
R  2 P2NAT3
*  *  BEGINNING OF FUNCTION 3 LAYOUT  *        **** *******
   3 PIWA2-FN3-ACCESS-KEY            A          21
   2 PIWA2-FN3-DUP-KEY-FLAG          A           1 /* OR FN3 CONTI PARITY
   2 PIWA2-FN3-LOCATION-STATUS       A           1
   2 PIWA2-FN3-COUNTY_COUNDARY       A           1
   2 PIWA2-FN3-PREFERRED-LGC1        A           2
   2 PIWA2-FN3-PREFERRED-LGC2        A           2
   2 PIWA2-FN3-PREFERRED-LGC3        A           2
   2 PIWA2-FN3-NUM-X-ST-LOW-END      A           1
   2 PIWA2-FN3-LOW-B5SC              A           6 (1:5) /* 30-BYTES
   2 PIWA2-FN3-NUM-X-ST-HI-END       A           1
   2 PIWA2-FN3-HI-B5SC               A           6 (1:5) /* 30-BYTES
   2 PIWA2-FN3-REVERSALFLAG          A           1
   2 PIWA2-FN3-LIONKEY               A          10
R  2 PIWA2-FN3-LIONKEY
   3 PIWA2-FN3-LION-BORO             A           1
   3 PIWA2-FN3-LION-FACECODE         A           4
   3 PIWA2-FN3-LION-SEQ              A           5
   2 PIWA2-FN3-GENREC-FLAG           A           1
   2 PIWA2-FN3-SEG-LEN               A           5
   2 PIWA2-FN3-SEG-SLOP              A           3
   2 PIWA2-FN3-SEG-ORIENT            A           1
   2 PIWA2-FN3-MARBLE-RIKERS-FLAG    A           1
   2 FILLER-600                      A          19
   2 PIWA2-FN3-SEGMENT-ID            A           7
   2 PIWA2-FN3-DOT-SLA               A           1
   2 PIWA2-FN3-CURVE-FLAG            A           1
   2 PIWA2-FN3-DOG-LEG               A           1
   2 PIWA2-FN3-FEATURE-TYPE          A           1
   2 PIWA2-FN3-SEGMENT-TYPE          A           1
   2 FILLER-700                      A           5
*  * PIWA2-FN3-LEFT-SIDE-OF-STR      *         **** *****
   2 PIWA2-FN3-L-COM-DIST            A           3
R  2 PIWA2-FN3-L-COM-DIST
   3 PIWA2-FN3-L-COM-DIST-BORO       A           1
   3 PIWA2-FN3-L-COM-DIST-NUM        A           2
   2 PIWA2-FN3-L-LOW-HOUSENUM        A          16 /* DISPLAY FORMAT
   2 PIWA2-FN3-L-HI-HOUSENUM         A          16 /* DISPLAY FORMAT
   2 FILLER-800                      A          32 /* FOR FUTURE USE
   2 FILLER-850                      A           1 /* FOR GSS USE
   2 PIWA2-FN3-L-ZIP                 A           5
   2 PIWA2-FN3-L-HEALTH-AREA         A           4
   2 PIWA2-FN3-L-POLICE-DIST         A           4
R  2 PIWA2-FN3-L-POLICE-DIST
   3 PIWA2-FN3-L-POL-PAT-BORO-CMD    A           1
   3 PIWA2-FN3-L-POL-PRECINCT        A           3
   2 PIWA2-FN3-L-FIRE-DIV            A           2
   2 PIWA2-FN3-L-FIRE-BAT            A           2
   2 PIWA2-FN3-L-FIRE-CO             A           4
R  2 PIWA2-FN3-L-FIRE-CO
   3 PIWA2-FN3-L-FIRE-CO-TYPE        A           1
   3 PIWA2-FN3-L-FIRE-CO-NUM         A           3
   2 PIWA2-FN3-L-SCHL-DIST           A           2
   2 PIWA2-FN3-L-DYN-BLK             A           3
   2 PIWA2-FN3-L-INSTRUCT-DIV        A           2
   2 FILLER-880                      A           7
   2 PIWA2-FN3-L-1990-CENS-TRCT      A           6
   2 PIWA2-FN3-L-2000-CENS-TRCT      A           6
   2 PIWA2-FN3-L-2000-CENS-BLK       A           4
   2 FILLER-890                      A           1 /* RESERVE FOR DCP
   2 FILLER-900                      A          30
*  * PIWA2-FN3-RIGHT-SIDE-OF-STR     *         **** *****
```

GEOLP23 COPY File (continued)

```
    2 PIWA2-FN3-R-COM-DIST              A          3
R   2 PIWA2-FN3-R-COM-DIST
    3 PIWA2-FN3-R-COM-DIST-BORO         A          1
    3 PIWA2-FN3-R-COM-DIST-NUM          A          2
    2 PIWA2-FN3-R-LOW-HOUSENUM          A         16 /* DISPLAY FORMAT
    2 PIWA2-FN3-R-HI-HOUSENUM           A         16 /* DISPLAY FORMAT
    2 FILLER-1000                       A         32 /* FOR FUTURE USE
    2 FILLER-1050                       A          1 /* FOR GSS USE
    2 PIWA2-FN3-R-ZIP                   A          5
    2 PIWA2-FN3-R-HEALTH-AREA           A          4
    2 PIWA2-FN3-R-POLICE-DIST           A          4
R   2 PIWA2-FN3-R-POLICE-DIST
    3 PIWA2-FN3-R-POL-PAT-BORO-CMD      A          1
    3 PIWA2-FN3-R-POL-PRECINCT          A          3
    2 PIWA2-FN3-R-FIRE-DIV              A          2
    2 PIWA2-FN3-R-FIRE-BAT              A          2
    2 PIWA2-FN3-R-FIRE-CO               A          4
R   2 PIWA2-FN3-R-FIRE-CO
    3 PIWA2-FN3-R-FIRE-CO-TYPE          A          1
    3 PIWA2-FN3-R-FIRE-CO-NUM           A          3
    2 PIWA2-FN3-R-SCHL-DIST             A          2
    2 PIWA2-FN3-R-DYN-BLK               A          3
    2 PIWA2-FN3-R-INSTRUCT-DIV          A          2
    2 FILLER-1080                       A          7
    2 PIWA2-FN3-R-1990-CENS-TRCT        A          6
    2 PIWA2-FN3-R-2000-CENS-TRCT        A          6
    2 PIWA2-FN3-R-2000-CENS-BLK         A          4
    2 FILLER-1090                       A          1 /* RESERVE FOR DCP
    2 FILLER-1100                       A         30
```

GEOLP21A COPY File

```
   1 GEOLP21A                                    /*FCT 1A,BL USE SAME WA2 LAYOUT
*  * THE FIELD P2NAT1A IS USED AS A PARAMETER TO CALL GEOSUPPORT
   2 P2NAT1A                       A          21
R  2 P2NAT1A
   3 PIWA2-1A-ACCESS-KEY           A          21
   2 PIWA2-1A-CONT-PARITY          A           1 /* OR DUP ADDR IND
   2 PIWA2-1A-LOW-HOUSENUM         A          11 /* SORT FORMAT
   2 PIWA2-1A-BBL                  A          10
R  2 PIWA2-1A-BBL
   3 PIWA2-1A-BBL-BORO             A           1
   3 PIWA2-1A-BLOCK                A           5
   3 PIWA2-1A-LOT                  A           4
   2 PIWA2-1A-LOT-VER0             A           1
   2 PIWA2-1A-SCC                  A           1
   2 FILLER-100                    A           1
   2 PIWA2-1A-GENERAL-LOT-INFO
   3 PIWA2-1A-RPAD-BLDG-CLASS      A           2
   3 PIWA2-1A-CORNER-CODE          A           2
   3 PIWA2-1A-NUM-OF-STRUCTURES    A           4
   3 PIWA2-1A-NUM-OF-BLOCKFACES    A           2
   3 PIWA2-1A-INTERIOR-FLAG        A           1
   3 PIWA2-1A-VACANT-FLAG          A           1
   3 PIWA2-1A-IRREG-LOT-FLAG       A           1
   2 PIWA2-1A-MARBLE-RICKERS-FLAG  A           1
   2 PIWA2-1A-ADDR-LIST-OVFLOW-FLAG A          1
   2 PIWA2-1A-STROLL-KEY           A          19
R  2 PIWA2-1A-STROLL-KEY
   3 PIWA2-1A-STROLL-BORO          A           1
   3 PIWA2-1A-STROLL-5SC           A           5
   3 PIWA2-1A-STROLL-SIDE-OF-STR   A           1 /* L OR R
   3 PIWA2-1A-STROLL-HI-HOUSENUM   A          11 /* SORT FORMAT
   3 FILLER-200                    A           1
   2 FILLER-300                    A           1 /* FOR GSS USE
   2 PIWA2-1A-BIN                  A           7
   2 PIWA2-1A-CONDO-FLAG           A           1
   2 FILLER-400                    A           1
   2 PIWA2-1A-RPAD-CONDO-ID-NUM    A           4
   2 PIWA2-1A-CONDO-UNIT-ID-NUM    A           7
   2 PIWA2-1A-CONDO-BILL-BBL       A          10
   2 PIWA2-1A-CONDO-BILL-BBL-VER   A           1
   2 PIWA2-1A-CONDO-BILL-BBL-SCC   A           1
   2 PIWA2-1A-CONDO-LOW-BBL        A          10
   2 PIWA2-1A-CONDO-LOW-BBL-VER    A           1
   2 PIWA2-1A-CONDO-HIGH-BBL       A          10
   2 PIWA2-1A-CONDO-HIGH-BBL-VER   A           1
   2 FILLER-500                    A          15
   2 PIWA1-1A-COOP-NUM             A           4
   2 PIWA2-1A-SANBORN              A           8
R  2 PIWA2-1A-SANBORN
   3 PIWA2-1A-SANBORN-BORO         A           1
   3 PIWA2-1A-SANBORN-VOL          A           3
   3 PIWA2-1A-SANBORN-PAGE         A           4
   2 PIWA2-1A-COMMERC-DIST         A           5
   2 PIWA2-1A-DOF-MAP-BORO         A           1
   2 PIWA2-1A-DOF-MAP-SECVOL       A           4
   2 PIWA2-1A-DOF-MAP-PAGE         A           4
   2 FILLER-1A-RESERVED-DCP        A          23
   2 PIWA2-1A-X-COORD              A           7
   2 PIWA2-1A-Y-COORD              A           7
   2 FILLER-650                    A          25
   2 FILLER-700                    A           8 /* FOR GSS USE
   2 PIWA-1A-NUM-OF-ADDR           A           4
   2 PIWA2-1A-ADDR-LIST                          (1:21)
   3 PIWA2-1A-LIST-LOW-HOUSENUM    A          16 /* DISPLAY FORMAT
   3 PIWA2-1A-LIST-HI-HOUSENUM     A          16 /* DISPLAY FORMAT
   3 PIWA2-1A-LIST-BORO            A           1
```

GEOLP21A COPY File (continued)

```
   3 PIWA2-1A-LIST-5SC                A           5
   3 PIWA2-1A-LIST-BIN                A           2
   3 PIWA2-1A-LIST-ADDR-TYPE          A           7
   3 PIWA2-1A-LIST-SIDE-OF-STR        A           1 /* L OR R
   3 PIWA2-1A-ADDR-TYPE               A           1 /* P=NAP, B=NAB, BLANK=NORMAL
   3 FILLER-800                       A           4
```

GEOLP2AL COPY File

```
    1 GEOLP2AL                                      /*FCT 1A,BL USE SAME LONG WA2
*   * THE FIELD P2NATAL IS USED AS A PARAMETER TO CALL GEOSUPPORT
    2 P2NATAL                       A          21
R   2 P2NATAL
    3 PIWA2-1AL-ACCESS-KEY          A          21
    2 PIWA2-1AL-CONT-PARITY         A           1 /* OR DUP ADDR IND
    2 PIWA2-1AL-LOW-HOUSENUM        A          11
    2 PIWA2-1AL-BBL                 A          10
R   2 PIWA2-1AL-BBL
    3 PIWA2-1AL-BBL-BORO            A           1
    3 PIWA2-1AL-BLOCK               A           5
    3 PIWA2-1AL-LOT                 A           4
    2 PIWA2-1AL-LOT-VER             A           1
    2 PIWA2-1AL-SCC                 A           1
    2 FILLER-100                    A           1
    2 PIWA2-1AL-GENERAL-LOT-INFO
    3 PIWA2-1AL-RPAD-BLDG-CLASS     A           2
    3 PIWA2-1AL-CORNER-CODE         A           2
    3 PIWA2-1AL-NUM-OF-STRUCTURES   A           4
    3 PIWA2-1AL-NUM-OF-BLOCKFACES   A           2
    3 PIWA2-1AL-INTERIOR-FLAG       A           1
    3 PIWA2-1AL-VACANT-FLAG         A           1
    3 PIWA2-1AL-IRREG-LOT-FLAG      A           1
    2 PIWA2-1AL-MARBLE-RIKERS-FLAG  A           1
    2 PIWA2-1AL-ADDR-LIST-OVFLOW-FLAG  A        1
    2 PIWA2-1AL-STROLL-KEY          A          19
R   2 PIWA2-1AL-STROLL-KEY
    3 PIWA2-1AL-STROLL-KEY-BORO     A           1
    3 PIWA2-1AL-STROLL-KEY-5SC      A           5
    3 PIWA2-1AL-STROLL-SIDE-OF-STR  A           1 /* L OR R
    3 PIWA2-1AL-STROLL-HI-HOUSENUM  A          11 /* SORT FORMAT
    3 FILLER-200                    A           1
    2 FILLER-300                    A           1 /* FOR GSS USE
    2 PIWA2-1AL-BIN                 A           7
    2 PIWA2-1AL-CONDO-FLAG          A           1
    2 FILLER-400                    A           1
    2 PIWA2-1AL-RPAD-CONDO-ID-INUM  A           4
    2 PIWA2-1AL-CONDO-UNIT-ID-NUM   A           7
    2 PIWA2-1AL-CONDO-BILL-BBL      A          10
    2 PIWA2-1AL-CONDO-BILL-BBL-VER  A           1
    2 PIWA2-1AL-CONDO-BILL-BBL-SCC  A           1
    2 PIWA2-1AL-CONDO-LOW-BBL       A          10
    2 PIWA2-1AL-CONDO-LOW-BBL-VER   A           1
    2 PIWA2-1AL-CONDO-HIGH-BBL      A          10
    2 PIWA2-1AL-CONDO-HIGH-BBL-VER  A           1
    2 FILLER-500                    A          15
    2 PIWA2-1AL-COOP-NUM            A           4
    2 PIWA2-1AL-SANBORN             A           8
R   2 PIWA2-1AL-SANBORN
    3 PIWA2-1AL-SANBORN-BORO        A           1
    3 PIWA2-1AL-SANBORN-VOL         A           3
    3 PIWA2-1AL-SANBORN-PAGE        A           4
    2 PIWA2-1AL-COMMERC-DIST        A           5
    2 PIWA2-1AL-DOF-MAP-BORO        A           1
    2 PIWA2-1AL-DOF-MAP-SECVOL      A           4
    2 PIWA2-1AL-DOF-MAP-PAGE        A           4
    2 FILLER-600                    A          23
    2 PIWA2-1AL-X-COORD             A           7
    2 PIWA2-1AL-Y-COORD             A           7
    2 FILLER-650                    A          25
    2 FILLER-700                    A           8 /* FOR GSS USE
    2 PIWA2-1AL-NUM-OF-BINS         A           4
    2 PIWA2-1AL-BINS                A           7 (1:2500)
```

GEOLP23S COPY File

```
  1 GEOLP23S
* * THE FIELD P2NAT3S IS USED AS A PARAMETER TO CALL GEOSUPPORT
  2 P2NAT3S                         A   21
R 2 P2NAT3S
  3 PIWA2-3S-ACCESS-KEY             A   21
R 3 PIWA2-3S-ACCESS-KEY
  4 FILLER-GSS                      A    2
  4 PIWA2-3S-PORS-STNAME-IND        A    1
  4 PIWA2-3S-BORO                   A    1 /* P=PRIMARY
  4 PIWA2-3S-5SC                    A    5 /* S=SECONDARY
  4 PIWA2-3S-LGC                    A    2 /* BLANK IF P IN
  4 FILLER                          A   10 /* POSITION 3
  2 PIWA2-3S-NUM-OF-INTERSECTS      A    3
  2 PIWA2-3S-LIST-OF-INTERSECTS            (1:350)
  3 PIWA2-3S-MARBLE-RICKERS-FLAG    A    1
  3 PIWA2-3S-DISTANCE               A    5
  3 PIWA2-3S-GAP-FLAG               A    1
  3 FILLER-100                      A    7
  3 PIWA2-3S-NUM-OF-STR             A    1
  3 PIWA2-3S-B7SC                   A    8 (1:5)
```

GLOSSARY OF TERMS AND ACRONYMS

Citations in brackets are references to sections of the UPG where the given term is defined or is principally discussed. Phrases in **bold typeface** have entries in this glossary.

ADDRESSABLE PLACE NAME [Section III.6]: A **place name** that can be combined with a house number to form an address. (Contrast with **non-addressable place names**.) Geosupport's **address-processing functions** accept addressable place names as input data for the specification of an address. Some Manhattan examples are PENN PLAZA, WASHINGTON SQUARE VILLAGE and NEW YORK PLAZA.

ADDRESS / INTERSECTION TO MAP ZONES (AIMZ) [Section I.1]: A Geosupport CICS utility transaction that allows the user to enter an address, **place name**, intersection, tax lot identifier, or **Building Identification Number** and receive back a screen display of a set of map identifiers corresponding to the input location. The use of AIMZ requires no programming skills and AIMZ is not documented in detail in this **UPG**.

ADDRESS-PROCESSING FUNCTION [Chapter V]: Any of the Geosupport **function**s that accept the input of addresses. Currently, these are Functions 1, 1A and 1E. All of the address-processing functions also accept **non-addressable place names** as input data with no input house numbers specified. The address-processing functions are a subset of the **location-processing functions**.

ALIAS [Section IV.2]: Two street names (or names of non-street geographic features) are aliases of each other if they are alternative names for the same street (or non-street feature) or any portion(s) thereof, or are spelling variants of the same street (or non-street feature) name. **Partial street names** are considered spelling variants, and therefore aliases, of the corresponding full street names. The alias relationship is embodied in the assignment of Geosupport **street codes**: two street names are aliases of each other if and only if they have the same borough-and-five-digit **street code**. Some examples of aliases in Manhattan: 6 AVENUE, SIXTH AVENUE, and AVENUE OF THE AMERICAS are all aliases of each other. SEVENTH AVENUE, 7 AVENUE, FASHION AVENUE and ADAM C POWELL JR BOULEVARD are all aliases of each other, even though some of these names are valid for differing portions of the street.

ALIASES (in GBAT) [Section IX.6]: User-defined street name aliases may be used in GBAT applications to supplement the set of street names that Geosupport recognizes. GBAT aliases are typically used to handle a consistent misspelling of a street name. The GBAT aliases are different from the **Alias**es described in Section IV.2.

AIMZ - see **Address / Intersection to Map Zones**

API - see **Geosupport Application Programming Interface**

BACKGROUND COMPONENT [Section I.5]: The component of the Geosupport System in which **GSS** updates and validates geographic base files from which new releases of the **foreground component** files are periodically generated. The background component software and files are not directly accessed by users.

BBL ('Borough/Block/Lot') [Section VI.2]: A unique identifier for a parcel of real property, or tax lot, in New York City. The BBL is a 10-byte item formed by concatenating the one-byte borough code, five-byte tax block number and four-bye tax lot number. The New York City Department of Finance assigns tax block and tax lot numbers.

BEND [Section III.6]: A **pseudo-street name** that Geosupport accepts as street name input to specify a bending point of a street. Geosupport treats a point along a street as a bending point if the angle of the street at that point is not within the range 160-200 degrees, that is, if it is not within 20 degrees of a straight line.

BILLING BBL [Section VI.4]: A special **BBL** assigned by the Department of Finance to each condominium, to enable identification of the condominium in its entirety as distinct from the condominium's individual units.

BIN - see **Building Identification Number**

BLOCK FACE [Section VII.3]: A continuous frontage of a physical city block along one street, encompassing any bending points of the street within that frontage.

BUILDING IDENTIFICATION NUMBER (BIN) [Section VI.3]: A unique, immutable identifier for each building in New York City. BINs are not to be confused with addresses. BINs are assigned by the Geographic Systems Section (**GSS**) at the Department of City Planning.

CHARACTER-ONLY WORK AREA (COW) [Appendix 12, Appendix 13 and Appendix 14]: The Geosupport work areas that have long been in use are called the Mainframe-Specific Work Areas (MSWs). Most of the MSWs contain one or more packed decimal fields, a data encoding schema unique to IBM mainframes. An alternative set of Geosupport work areas was introduced in 2002. It is called the Character-Only Work Areas (COWs) which, as the name implies, contain character fields only. The COW is an essential part of a long-term effort to port the Geosupport System to other platforms. From now on, all new applications should be designed to use the COWs only. We also recommend that all existing applications be converted to use the COWs. See also Glossary entry for **Work Areas**.

CITY LIMIT [Section III.6]: A **pseudo-street name** that Geosupport accepts as street name input to refer to locations on the Bronx-Westchester County and Queens-Nassau County borders.

COMPACT FORMAT [Section III.3]: A Geosupport format for **normalize**d geographic feature names. The compact format is suitable for display but not for sorting. Contrast with the **sort format**, which is suitable for sorting but not for display.

COMPLEX [Section III.6]: A group of related buildings and/or other geographic features at one site. The name of a complex is a **NAP** (Non-Addressable Place Name). Examples of complexes include housing projects, university and hospital campuses, cultural complexes (such as Lincoln Center) and airports. Compare to **simplex** and **constituent entity of a complex**.

COMPUTER SERVICE CENTER (CSC): A data center operated by the New York City Department of Information Technology and Telecommunications (DoITT) as a service to all agencies of the City of New York. Many city agencies run computer applications on CSC's IBM mainframe, which is located in Brooklyn. The Geosupport System is installed at CSC and at several other city data centers (see Appendix 7). **GSS** conducts Geosupport software development and testing and some Geosupport data file maintenance and generation on the CSC mainframe.

CONSTITUENT ENTITY OF A COMPLEX [Section III.6]: An individual building or other geographically identifiable feature that is part of a **complex**. Examples are the buildings in Lincoln Center and in Stuyvesant Town.

COPY LIBRARY, COPY FILES [Section VIII.4]: Many programming languages have a facility for accessing external files of source code called COPY files during application program compilation. COPY files reside in a partitioned data set (PDS) called a COPY library. The Geosupport System has COPY libraries containing source code layouts of the **work areas** in Assembler, PL/1, COBOL, C and NATURAL. The use of the Geosupport COPY libraries by application developers is optional but is strongly recommended.

COW - See **Character-Only Work Area**

CSC - see **Computer Service Center**

DAPS - see **Duplicate Address Pseudo-Street Name**

DEAD END [Section III.6]: A **pseudo-street name** that Geosupport accepts as street name input to refer to a termination point of a street at which there are no cross streets.

DEPARTMENT OF INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS (DoITT): An agency of the City of New York responsible for city government-wide information technology infrastructure support. DoITT operates the **Computer Service Center**.

DISPLAY FUNCTION [Sections IV.6 and V.2]: Any of the Geosupport **function**s that provide data items that can be used to display geographic locations on application screens, reports, mailing labels etc. Specifically, the display **function**s provide street names corresponding to input **street code**s, and provide house numbers in **HND** format corresponding to input house numbers in **HNI** (MSW) or **HNS** (COW) format. Note that the display **function**s do not actually display anything themselves; they merely provide data items that are suitable for an application to display. Currently, the display **function**s are Functions D, DG and DN.

DoITT - see **Department of Information Technology and Telecommunications**

DRIVER, GEOSUPPORT [Section II.1]: A Geosupport load module that serves as an interface enabling application programs to access Geosupport via **API** calls. There are two different drivers, one for batch applications and one for CICS applications. Application developers must link-edit the appropriate driver into the application program.

DUPLICATE ADDRESS PSEUDO-STREET NAME (DAPS) [Section V.6]: A **pseudo-street name** accepted as street name input by Geosupport in duplicate address situations. DAPSs enable applications to specify which instance of a duplicated address the application wishes to process.

FOREGROUND COMPONENT [Section I.5]: The component of the Geosupport System that is directly accessed by a user application via the **API**. The foreground component includes both software and files.

FREE-FORM ADDRESS [Section V.3]: An address expressed with the house number and street name stored together in a single field. (Compare with **parsed-form address**.) Geosupport can process free-form addresses in which the house number and street name are passed together in the **WA1** input street name field (and no value is passed in the separate **WA1** input house number field).

FUNCTION [Sections I.2, I.4]: The Geosupport System is organized into more than a dozen distinct functions that can be accessed by the user. Each function is identified by a one- or two-character function code.

GBAT - see **Geosupport Batch Address Translator**

GEOCODE [Section I.2]: The process of associating higher-level geographic information, such as the police precinct, zip code or census tract, with a specific geographic location, such as an address or street intersection. Geocoding is one of the Geosupport System's most important services.

GEOGRAPHIC RETRIEVAL CONSISTENCY [Section I.3]: Retrieval of information by geographic location in a manner that is independent of how the location is specified. The ability of an application to retrieve data consistently by geographic location from the application's own files is a critical design issue for many applications. One important means of implementing geographic retrieval consistency in an application is to use B5SCs (see the entry for **alias**) instead of street names in the retrieval key.

GEOSUPPORT APPLICATION PROGRAMMING INTERFACE (API) [Section II.1]: The Geosupport facility that enables user-written application programs to interact with Geosupport via standardized program calls. The API involves the use of a Geosupport **driver** module and Geosupport **work areas**.

GEOSUPPORT BATCH ADDRESS TRANSLATOR (GBAT) [Section IX.1]: The Geosupport System's batch utility program.

GEOSUPPORT ONLINE ADDRESS TRANSLATOR (GOAT) [Section I.1]: The Geosupport System's principal CICS utility transaction. GOAT is an inquiry transaction that allows the user to request any Geosupport **function**, enter input data and receive back a formatted screen display of the corresponding output information provided by that **function**. The use of GOAT requires no programming skills and it is not documented in detail in this **UPG**.

GEOSUPPORT RETURN CODE (GRC) [Section II.2]: A two-byte code that is returned in **WA1** upon completion of every **API** call to Geosupport, indicating to the calling application the outcome of the call. (Not to be confused with operating system return codes or condition codes.) A GRC value of '00' signifies an unconditionally successful call. A GRC value of '01' signifies a **warning**. A GRC value of other than '00' or '01' signifies a **reject**. See also the Glossary entries for **Reason Code** and **Message**. See Appendix 4 for a comprehensive list of **GRC**s, **Reason Codes** and **Messages**.

GEOSUPPORT SYSTEM ADMINISTRATOR [Section I.1]: A designated staff member (generally a systems programmer) of a computer center where Geosupport is installed on a mainframe, responsible for installing new Geosupport file releases and software versions, and for trouble-shooting system-related Geosupport problems. Note: the Geosupport System Administrator is not necessarily responsible for providing application-related support to users.

GOAT - see **Geosupport Online Address Translator**

GRC - see **Geosupport Return Code**

GSS [Section I.1]: The Geographic Systems Section of the City of New York Department of City Planning's Information Technology Division. **GSS** is the developer and custodian of the Geosupport System.

HND - see **House Number in Display Format**

HNI - see **House Number in Internal Format**

HNS - see **House Number in Sort Format**

HOUSE NUMBER IN DISPLAY FORMAT (HND) [Section V.2]: One of Geosupport's three output **normalize**d house number formats. The HND is a format suitable for applications to use for display on screens, reports and mailing labels.

HOUSE NUMBER IN INTERNAL FORMAT (HNI) [Section V.2]: One of Geosupport's three output **normalize**d house number formats. The HNI is not suitable for display, because it is partly in packed decimal form, and it contains a code representing the house number suffix (if any) rather than the suffix itself. The HNI is used internally in the Geosupport System, and it is not of direct significance to most applications. HNI is valid in MSW only.

HOUSE NUMBER IN SORT FORMAT (HNS) [Section V.2]: One of Geosupport's three output **normalize**d house number formats. The HNS is not suitable for display, because it has an internal format and contains a code representing the house number suffix (if any) rather than the suffix itself. The HNS is used internally in the Geosupport System, and it is not of direct significance to most applications. HNS is valid in COW only.

ID-PROCESSING FUNCTION [Section I.4]: Any **location-processing function** that processes identification codes. Currently, the ID-processing functions are Function BL, which processes tax lots specified by an input **BBL**; and Function BN, which processes buildings specified by an input **BIN**

INPUT FIELD (IN A WORK AREA) [Section II.3]: A field into which the user application inserts a value to be passed to Geosupport. See also **output field**, **WA1** and **WA2**. **WA1** has both input and output fields. **WA2** has output fields only.

LDF- see **LION Differences File**

LGC - see **Local Group Code**

LION DIFFERENCES FILE (LDF): The LION Differences File (LDF) is a sequential file containing records documenting certain types of changes that have occurred between a particular release of LION and the immediately previous LION release. A new LDF 'edition' is 'published' in conjunction with each new production release of LION. The changes documented in the LDF relate to node changes and segment changes.

LION FILE [Section VII.1]: A **background component** file that is a digital map of New York City. LION contains a single-line representation of the city's streets and non-street features. Geosupport's **street configuration** processing is based on that representation.

LOCAL GROUP CODE (LGC) [Section IV.5]: The LGC consists of the sixth and seventh digits of the ten-digit **street code**. The LGC corresponds to a set of **locally valid street names** for the given street.

LOCALLY VALID STREET NAME [Section IV.5]: A name of a street that is valid for a particular portion (possibly all) of the street. The set of street names that are valid for the same portion of a street constitute a 'local group' and share the same **LGC** value.

LOCATION-PROCESSING FUNCTION: Any of the Geosupport **function**s that accept the input of a geographic location. These can be sub-classified into the **address-processing functions** (Functions 1, 1A and 1E); the **street-configuration-processing functions** (Functions 2, 3, 3C and 3S); and the **ID-processing functions** (Functions BL and BN).

MAINFRAME-SPECIFIC WORK AREA (MSW (a.k.a. MFS)) - see **Character-Only Work Area**

MESSAGE [Section II.2]: A **WA1** output item returned for all **warnings** and **rejects**, consisting of an appropriate explanatory text message. See Appendix 4 for a comprehensive list of **GRC**s, **Reason Codes** and **Messages**.

MFS - see **MSW**

MSW - see **Mainframe-Specific Work Area**

NAP - see **Non-addressable Place Name**

NAUB - see **Non-addressable Un-named Building**

NODE [Section VII.2]: Either a conventional intersection of a street with another street, or a **pseudo-intersection** of a street with a **pseudo-street**.

NON-ADDRESSABLE PLACE NAME (NAP) [Section III.6]: A place name that cannot be combined with a house number to form an address. Examples: CITY HALL, EMPIRE STATE BUILDING, PLAZA HOTEL, LINCOLN CENTER, LA GUARDIA AIRPORT. A NAP can either be the name of a **simplex**, a **complex**, or a **constituent entity of a complex**. Geosupport's **address-processing functions** accept many NAPs as input data.

NON-ADDRESSABLE UN-NAMED BUILDING (NAUB) [Section VI.3]: A building that has neither addresses nor **NAP**s, and can only be identified by its **BIN**. Typical example is a storage shed on the grounds of an industrial property.

NORMALIZE [Section III.2 for street names, Section V.2 for house numbers]: To produce a version of a data item in a standardized format. Geosupport normalizes every input geographic feature name into one of two formats selected by the user application, called the **compact format** and the **sort format**. Geosupport also normalizes every input house number. Geosupport returns output normalized names and house numbers to the calling application in **WA1**.

OUT-OF-SEQUENCE ADDRESS [Section V.10]: An address such that the house number is out of sequence relative to nearby house numbers along the given street. For an input out-of-sequence address, the output information that Functions 1 and 1E return is based on the street segment where the out-of-sequence address is actually located, including the cross streets and geographic district identifiers. The Spatial Coordinates returned are those of a point calculated under the assumption that the building entrance is located at the midpoint of the block face. A warning is issued for any address on a block face containing an out-of-sequence address.

OUTPUT FIELD (IN A WORK AREA) [Section II.3]: A field into which Geosupport inserts a value to be returned to the calling user application. See also **input field**, **WA1** and **WA2**. **WA1** has both input and output fields. **WA2** has output fields only.

PARSED-FORM ADDRESS [Section V.3]: An address that is expressed with the house number and street (name or code) stored in separate fields. (Compare to **free-form address**.)

PARTIAL STREET NAME [Section III.4]: A street name formed from a full normalized street name by deleting one or more entire words from the end of the full street name. For example, in Manhattan, READE is a partial street name for READE STREET. Geosupport accepts a partial street name as an input street name when the partial street name unambiguously represents a unique full street name in the specified input borough.

PLACE NAME [Section III.6]: A name of a geographic feature other than a street name or a **pseudo-street name**. Examples of place names are the names of building complexes (such as university campuses, housing projects, hospital campuses etc.), individual named buildings (such as CITY HALL, EMPIRE STATE BUILDING, museums, hotels, theaters, stadiums etc.), parks, islands, airports etc. Geosupport recognizes some New York City place names, and more are being added over time. There are several types of place names; see Glossary entries for **Addressable Place Name, Non-Addressable Place Name, Simplex, Complex and Constituent Entity of a Complex.**

PREFERRED STREET NAME [Section IV.5]: If more than one local group of street names is valid at a particular location along a street, **GSS** designates one of them as the 'preferred' local group for that location. The preferred street name is the **principal street name** of the preferred local group.

PRIMARY STREET NAME [Section IV.3]: For every street in NYC, that is, for every valid B5SC value, **GSS** designates one spelling of one name of the street as the primary street name. Function D can be used to obtain the primary street name for a given B5SC value.

PRIMING WA1 [Section II.3]: The part of the API procedure in which the calling application program inserts values into **WA1 input fields** in preparation for issuing a call to the **driver**. Priming WA1 is how an application requests the **function** to be performed, passes the input geographic data (such as an address) to be processed, and specifies processing options.

PRINCIPAL STREET NAME OF LOCAL GROUP [Section IV.5]: The street name that **GSS** has designated as the 'best' representative from among all the names in a local group. Function DG can be used to obtain the principal street name for a given B7SC value.

PSEUDO-ADDRESS [Section VI.5]: An address unofficially assigned by **GSS** to a street frontage of a tax lot that has no 'real' building addresses, such as a driveway. Function 1A accepts pseudo-addresses as input.

PSEUDO-INTERSECTION [Section VII.2]: A point along a street specified in terms of a

pseudo-street name, i.e., a bend, a dead end or a city limit point.

PSEUDO-STREET NAME [Section III.6]: An 'unofficial' street name that Geosupport accepts as street name input for certain geographic situations. **DAPS**s are pseudo-street names that the **address-processing functions** accept as input only for the city's very few cases of duplicate addresses (see Section V.6). DEAD END, CITY LIMIT, BEND and their **aliases** are pseudo-street names accepted as input by the **functions** that process **street configurations** (see Chapter VII).

REASON CODE [Section II.2]: A one-byte output **WA1** item that qualifies the reason for a **warning** or **rejection** with greater specificity than does the **GRC** alone. Non-blank reason codes are returned for all **warnings** and for selected **rejects**. See Appendix 4 for a comprehensive list of **GRC**s, **Reason Codes** and **Messages**.

REJECT, REJECTION [Section II.2]: An unsuccessful outcome of an **API** call to Geosupport, indicated by a **GRC** value other than '00' or '01', accompanied by an appropriate **Message**, and for selected rejects, by a **Reason Code**.

RELEASE (OF GEOSUPPORT FOREGROUND FILES) [Section I.5]: Geosupport's **foreground component** files are read-only files, and are periodically replaced by updated files. Every foreground file is identified as belonging to a specific Geosupport release.

RESYNCHRONIZATION OF STREET CODES [Section IV.4]: The updating of Geosupport **street codes** stored in a user application file to reflect street code assignment changes made in a Geosupport **release**.

ROADBED [Section V.5]: A roadbed is a street segment that is bounded on both sides by a physical separator such as a sidewalk, median barrier or median strip. Street segments that have painted medians separating travel direction do not form multiple roadbeds. Well-known examples of streets with multiple roadbeds include Park Avenue in Manhattan, Queens Blvd in Queens and Ocean Parkway in Brooklyn.

SIMILAR NAME [Section III.5]: When an input street name is rejected, Geosupport returns a list of up to ten 'similar names' in **WA1**, as an aid to the application in handling the **reject**. A 'similar name' is a valid full street name from the specified input borough that Geosupport, in accordance with certain criteria, deems to be similar to the rejected input street name.

SIMPLEX [Section III.6]: A 'stand-alone' named geographic feature, that is, a feature that has a **NAP** and that is not a **complex** or a **constituent entity of a complex**. Examples: Empire State Building, Plaza Hotel, Gramercy Park.

SNC - see **Street Name Code**

SNL - see **Street Name Normalization Length Limit**

SORT FORMAT [Section III.3]: A Geosupport format for **normalize**d geographic feature names. The sort format is suitable for sorting but not for display. Contrast with the **compact format**, which is suitable for display but not for sorting.

STREET CODE [Chapter IV]: In the Geosupport System, a set of numeric street codes is assigned to represent the city's street names and other geographic feature names. A borough code combined with a ten-digit street code, or B10SC, corresponds to a specific spelling of a specific street name in the given borough. Portions of the B10SC also have special significance. In particular, the first six bytes of the B10SC, the borough-and-five-digit street code (B5SC), encodes the **alias** relationship between street names.

STREET CONFIGURATION [Section VII.1]: A geographic location specified in terms of a combination of two or three streets. Street configurations include intersections, street segments, **block faces** and street stretches.

STREET-CONFIGURATION-PROCESSING FUNCTION [Chapter VII]: Any of the Geosupport **location-processing functions** that process **street configurations**. Currently, these are Function 2, which processes street intersections; Function 3, which processes street segments; Function 3C, which processes **block faces**; and Function 3S, which processes street stretches.

STREET NAME CODE (SNC): The final three digits of the B10SC (Borough and Ten-digit **Street Code**) are called the Street Name Code (SNC). Thus, the B10SC consists of the concatenation of the borough code, 5SC, **LGC** and SNC. The SNC serves simply to serialize the street names within a local group, so that the full B10SC is unique to a specific spelling of a specific street name.

STREET NAME NORMALIZATION LENGTH LIMIT (SNL) [Section III.2]: A user-specifiable parameter that sets the maximum length in bytes within which Geosupport **normalizes** input street names. The default value is 32.

UPG - see **User Programming Guide**

USER PROGRAMMING GUIDE (UPG) [Section I.6]: This document.

VANITY ADDRESS [Sections V.9]: An address such that the street name refers to a different street than the one on which the referenced building entrance is actually located. For an input vanity address, the output information that Functions 1 and 1E return is based on the street segment where the vanity address is actually located, including the cross streets, geographic district identifiers and spatial coordinates. A warning is issued accordingly.

The output information that Function 1A returns is based on the building associated with the vanity address. No warning is issued for Function 1A.

VERSION (OF GEOSUPPORT FOREGROUND SOFTWARE) [Section I.5]: Self-explanatory. Contrast use of the term 'version' for Geosupport software and '**release**' for Geosupport data files.

VESTIGIAL FEATURE [Section I.5]: An element of the Geosupport System, such as a **function**, a **work area,** a data item or a JCL statement, that is obsolete and has been superseded by an enhancement. Vestigial features may continue to be operational but should not be used in new applications, and should be eliminated from existing ones.

WARNING [Section II.2]: A conditionally successful completion of an **API** call to Geosupport. A warning is signified by a GRC value of '01' and an accompanying **Reason Code** and **Message**. In most cases, it is appropriate for applications to treat warnings in the same way as successful completions.

WA1, WA2 - see **Work Areas**

WORK AREAS [Section II.1]: Standard-layout blocks of data in memory that are shared between Geosupport and an application. The Geosupport work areas are an essential component of the Geosupport **API**, and constitute the sole means by which information passes between the application and Geosupport. Different Geosupport **functions** use different work area layouts. **API** calls can involve the passing of either one work area, called **Work Area 1 (WA1)**, or two work areas, **WA1** and **Work Area 2 (WA2)**.