Data Builds

Reference Manual

Preliminary Verision 2.0

Sagent Technology, Inc.

Data Builds Reference Manual

Manual Version: 0600

Trademark Notices:

ZIP Code, ZIP+4 are registered trademarks of the U.S. Postal Service.

Within this manual, other trademarked names may be used. Rather than putting a trademark symbol in every occurrence of a trademarked name, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intent of infringement on the Trademark.

USPS Notice:

Sagent Technology, Inc. holds a non-exclusive license to publish and sell ZIP+4 databases on optical and magnetic media. The price of the Sagent product is neither established, controlled, nor approved by the U.S. Postal Service.

Sagent Technology, Inc.

Corporate Office

800 W. El Camino Real, Suite 300
Mountain View, CA 94040
Phone: 650-493-7100
Fax: 650-815-3500

Development Office

4750 Walnut Street, Suite 200
Boulder, CO 80301
Phone: 303-442-2838
Fax: 303-440-3523

Contents

Chapter 1

Data files overview

GeoStan data files

Explanation

The following table describes the set of files that are used to support GeoStan.

File	Purpose
ctyst.dir	Required for the initialization of the Geostan Library. Included in the primary data set. CTYST.DIR contains information on city names and their corresponding ZIP codes. This allows tracking the ZIP codes associated with specific cities and quickly and accurately converting between them.
msac.dir	Required for the initialization of the Geostan Library. Included in the primary data set. MSAC.DIR contains cross reference information for all Metropolitan Statistical Areas. This allows one to reference information in multiple or nearby MSAs quickly.
parse.dir	Required for the initialization of the Geostan Library. Included in the primary data set. PARSE.DIR contains known variations of words that may be used as input or standard address information. This allows one to better understand all possible variations on input information and match to the standard or preferred information. Typically, parse.dir is created once each year during CASS certification and is not modified again until the next CASS cycle.
us.epi	Included in the Extended Performance Index Data Set. Extended Performance Index Data (EPI) consists of one file, US.EPI, which contains index information for the U.S. US.EPI contains index information to enhance the performance of the GeoStan library functionality by enabling additional index access into the GeoStan Data file US.GSD. The Extended Performance Index Data CD is a special order CD, not included with the standard Data Products Package. The Extended Performance Index data requires additional licensing from Sagent and is shipped to licensed customers only.
us.gsd	One of the Geostan Directory (GSD) files used to ascertain the correct matching address. Included in the primary data set. US.GSD is a required file for Geostan Library initialization.

File	Purpose
us.gsi	Included in the StateWide Intersection Index Data Set. US.GSI contains indexes that allow the GeoStan library to quickly find multiple intersections within a state. This information can be used to find all intersections in a specific state that match the search criteria entered by the user. The StateWide Intersection Index Data CD (Disc 4) is a special order CD, not included with the standard Data Products Package. Use of the StateWide Intersection Index data requires additional licensing from Sagent and is shipped to licensed customers only.
us.gsl	
us.z9	Included in the Primary Data Set and Spatial Data Set. Required for initialization of the Geostan Library. Not required for the initialization of the Spatial+ Library. The ZIP+4 centroid information is contained in the US.Z9 file. This file is used by the Geostan Library to assign ZIP+4 level geocodes and by the Spatial+ Library to extract ZIP+4 information.
us_p.gsi	Built as part of the merge process for us.gsd.
us_ps.gsi	Built as part of the merge process for us.gsd.
ust.gsd	One of the Geostan Directory (GSD) files used to ascertain the correct matching address. Included in the supplemental data set. UST.GSD is a supplemental file that provides data for street segments that do not contain addresses and is not required for Geostan Library initialization. The use of UST.GSD will not increase address standardization levels. UST.GSD is particularly useful when attempting to match and geocode intersections of rural roads, State, U.S. and Interstate highways.

Why the same files are built using different data sets

Each of the different providers make their data differently, so certain data sets are better than others for certain geographic areas. For example, NAVTECH data is better for metropolitan areas. GDT data is best for overall data use for the United States. We provide data files to the customer that are built with the data sets that best fill their needs. We have agreements with TIGER and GDT such that these are the only data sets we build on a monthly basis. If the customer wants data files built from other data sources, they must purchase the data themselves directly from the vendor and provide the data to us to build from.

Spatial+ data files

Spatial+ uses spatial boundary files that have an .gsb extension. They may either be point, line, or boundary files. We build and distribute three files to customers:

- states.gsb
- counties.gsb
- us.gsb

These files are created using a tool such as ArcView or Atlas GIS to build the intermediate mif/mid or BNA files. These intermediate files are then imported into Spatial through Desktop or a Spatial application, and the .gsb files are then output.

Files/utilities needed to do all data builds from scratch

Required Files

NOTE: *verMMYY means the current month's version of the build utilities (i.e. ver0400 for April 2000)*

United States Postal Service (USPS) files

USPS files are received on a monthly basis. These are released by the USPS on the 15th of the month and received by Sagent approximately 2 to 7 days later. The April 15th release of USPS data becomes the May release of Sagent Data products. The USPS also releases a "transaction file" around the 1st of each month which is intended to contain all of the transaction changes that will be the standard USPS release on the 15th of each month. We build each month's data with this transaction file to get the processes rolling. When the Standard release file from the USPS is received we perform some verification steps to ensure the Transaction file build and the Standard file build would create the same output.

- ZIP+4 files
- ZIP+4 Transaction file
- ZIP+4 Change file
- City/State file
- Line of Travel file

Geographic Data Technology (GDT) files

GDT files are received on a quarterly basis, and they contain the GDT raw street segment data. We receive GDT data in TIGER format; see the document "TIGER/Line Census Files, 1992", produced by the U.S. Department of Commerce, for a description of the TIGER data format. This document is kept by the Director of Data Products.

TIGER files

TIGER stands for Topographical Integrated Geographical Encoding Reference System; these files are received once a year, and contain the TIGER raw street segment files.

Required utilities

- Located in the GSDLOAD directory under each month's version (e.g. ver0400), sub-directories included.
- The contents of these directories could change, I am just listing what is currently used and where they exist on the hard drive.
- Some of the files in the parent build directory will be duplicated in sub-directories. This is because we copy certain directories only during different steps in the build process. Some of the files will have the same names but will be from different dates. It is very important to keep these files together since a specific utility might have been built using the set of libraries that are contained in that directory. If another library is used, the utility may not work.

- You can type the name of the utility on the command line, and a usage statement will be given including usage, valid flags, etc.

Utilities used in the build process

The Core Library and Data Build groups are responsible for all of the following applications:

Application	Purpose
Alias.exe	Creates street alias list
BatchInd.exe	Builds the .gsx files used for mapping
Building.exe	Creates high-rise records (a.k.a. building records)
Ctyst.exe	Builds the CTYST.DIR file
Finbuild.exe	Builds the finmbr.dat file
Gsdmerge.exe	Merge the state .gsd files into the US.GSD and UST.GSD files
GSLBuild.exe	Builds the US.GSL file
Indexbuild.exe	Builds the US.EPI file
Ltravel.exe	Exact purpose unknown; it is related to Line-of-Travel in GeoStan.
Msabuild.exe	Builds the MSAC.DIR file
Pass1.exe	Takes GDT data as input and standardizes it
Pass15.exe	
Pass2.exe	Chains range records together to build streets; this output is used by Pass3.exe
Pass3.exe	Creates state .gsd files. Unlike the other Pass utilities, each data file is processed twice using Pass3.exe
Pass4.exe	
Spalias.exe	Builds the US_P.GSI and US_PS.GSI files
SpIntBld.exe	Builds the US.GSI file
UniqueZ4.exe	
Z9dump.exe	Takes the US.Z9 and splits it up into ten .dbf files. Each file corresponds to the first digit of any possible ZIP code (0-9)
Z9xtract.exe	This is an older program that extracts the Z9 file back to a text file. It is no longer used.
Zip4load.exe	Builds the US.Z9 file
Zipcent.exe	Produces the data type centroid files
Zipindex.exe	Indexes the US.Z9

Build machine directories

The following tables list, for each directory:

- The resident applications

- The library files required by the applications
- The input data files used by the applications

The master directory is located on middleman’s S:\ drive.

gsdload\verMMYY

Application	Library	Data
Pass1.exe	Geostn32.dll (7/27/99)	Ctyst.dir (built every month)
Pass15.exe	Msvcrtd.dll (6/14/96)	Msac.dir
Pass2.exe	Nadlib.dll (2/11/97)	Parse.dir
Pass3.exe	Pipdll.dll (3/15/98)	Zipcode.gsb
Pass4.exe		Alaska.las
Alias.exe		Alaska.los
Building.exe		Conus.las
Ctyst.exe		Conus.los
Gsdmerge.exe		Hawaii.las
GSLBuild.exe		Hawaii.los
Ltravel.exe		Prvi.las
Msabuild.exe		Prvi.los
Spalias.exe		Geostan.lic (3/14/97)
SpIntBld.exe		Spatial.lic (3/14/97)
UniqueZ4.exe		
Z9dump.exe		
Z9xtract.exe		
Zip4load.exe		
Zipcent.exe		
Zipindex.exe		

gsdload\verMMYY\ctyst

Application	Library	Data
Ctyst.exe (copy of .exe in parent directory)		

gsdload\verMMYY\ext

Application	Library	Data
Spalias.exe	Geostn32.dll (6/9/99)	Geostan.lic (1/12/00)
SpIntBld.exe	Mapper.dll (5/28/98)	Spatial.lic (3/14/97)
	Msvcrtd.dll (6/14/96)	Mkem.bat
	Nadlib.dll (2/11/97)	Runalias.bat
	Pipdll.dll (3/15/98)	

gsdload\verMMYY\finmbr

Application	Library	Data
Finbuild.exe	Geostn32.dll (6/9/99)	Geostan.lic (1/12/00)
	Mapper.dll (11/9/98)	

gsdload\verMMYY\indexbuild

Application	Library	Data
Indexbuild.exe	Geostn32.dll (3/7/00)	Geostan.lic (3/14/97)

gsdload\verMMYY\usgsd

Application	Library	Data
Alias.exe	Geostn32.dll (6/25/99)	Geostan.lic (1/12/00)
BatchInd.exe	Pipdll.dll (3/15/98)	Mkem.bat
Building.exe		Uniq.bat
Gsdmerge.exe		
UniqueZ4.exe		

Data build hardware

All of the build machines are interchangeable in function; it doesn’t matter what files are built on what machine. Tasks are split up among all the build machines to speed up processing, but there are no set machines for certain jobs.

Build Machine	Notes
dogwood	
flamingo	
hog	
iguana	
middleman	S:\ drive is one of three core drives used for building
nautique	
penelope	
pinktotheend	
thistle	G:\ drive is one of three core drives used for building
first_data	Was really only used to build the NHDB Has the 500 GB storage array attached to it
lizard	Has tape library and backup tape drive attached to it Really only used in the past for backing up and unloading tapes Can be used for any build process if need be Z:\ drive is one of three core drives used for building

Build order

The data files must be built in the following order:

1. ctyst.dir
2. msac.dir (does not need to be built again if there are no changes from the prior build)
3. parse.dir (does not need to be built again if there are no changes from the prior build)
4. (usps only) us.gsd
5. (any other data type) us.gsd
6. (any other data type) ust.gsd
7. (any other data type) us_p.gsi
8. (any other data type) us_ps.gsi
9. (any other data type) us.gsi
10. (any other data type) us.z9
11. us.gsl
12. (GDT only) us.epi

Data build schedule

The following is a complete list of the data types we build with:

Data Type	When Files of this Data Type Are Built
Etak	Built once a year; the actual build date is determined by the client
GDTblack	Built only for standard release months
GDTred	Built and shipped to someone every month
Navtech	Until now, these files have been built once a year; they are becoming more popular and may start being built every other month
Tiger	Built only for standard release months
USPS	Built and shipped to someone every month

The following is a list of the custom data files we distribute:

Customer	Files Built for Customer
Allstate	ZipLastMM.txt ZipMM.txt Except.txt
Ameritech Advertising	Uszip7.txt
Dontech	9ZipIA.dbf 9ZipIL.dbf 9zipIN.dbf
Experian	Zip4cnt.dbf Zip4cnt.txt Ctyst.dir US.z9 (Std Tiger)
Progressive	Zip9.zip

The data needs to be built and ready for test by the 12th of the month prior to the month it will ship in.

For standard release months (January, March, May, July, September, and November) the data must ship to our replication company (currently ADS in Minneapolis, MN) on or about the 6th of the month. This allows them to replicate the data and deliver it to Florida on or about the 21st of the month. Florida then ships the data to our customers, which the customers should receive before the 28th.

For non-Standard months (February, April, June, August, October, and December) the data ships directly to each customer on the 20th.

Chapter 2

Data build procedures

Data files built from USPS data

ctyst.dir

Relationships/dependencies upon other files

This is the first file that needs to be built every month before any other step is started.

There must be a master build (e.g. \gsdload) directory on a network server that all the build machines can access in order to copy the needed utilities locally.

Pre-build procedure

These steps ensure that the same set of utilities and files are used to build the current month's data.

Note: *verMMYY means the current month's version of the build utilities (e.g. ver0400 for April 2000)*

1. Create a directory on the master build drive, under the GSDLOAD directory for the current month's build (i.e. VERMMYY, where MMYY stands for the current 2-digit month and 2-digit year, 0400).
2. Copy the contents including sub-directories of the previous month's VERMMYY directory.
3. Replace any utilities or files that may have changed since the previous month's build. The developers will let you know if there is a new version of the utility.
4. Unload ctystate.txt (City/State file), z4trans.txt (ZIP+4 Transaction file), z4change.txt (ZIP+4 change file) from USPS tapes. Unload the above files to a directory were you can access them from any of the build computers. Normally, `\\thistle\thistle_g\zipMMYY`. MMYY is the current 2-digit month and 2-digit year.
5. Unload ZIP+4 files from CD for current month, if you have received them, or from previous month's CDs. Unload the ZIP+4 files to:

   ```
   \\lizard\D_lizard\uspsraw\zip4
   ```

Build procedure

1. Create directory on the local drive for building the CTYST.DIR, normally named ctyst.
2. Local drive should have at least 1 GB of free space.
3. Copy ctystate.txt file to ctyst directory on local drive.
4. Copy contents of ctyst directory under `\gsdload\verMMYY` directory to ctyst directory on local drive.
5. Type the following at a command prompt:

   ```
   \>set GSDRELEASEDATE=20000415
   ```

   where 2000 is for the year, 04 is for the month of the data release and 15 is for the day (the day is always the 15th for all data files—this sets the beginning day of use for the data file and determines when the data file will expire.
6. One of the files copied from the ctyst directory is ctyst.exe. Run the following from the command prompt:

   ```
   \>ctyst ctystate.txt CTYST.DIR
   ```
7. The output from this will be the CTYST.DIR file. If you type the follow command at the prompt (in the CTYST.DIR build directory) it will give you the copyright information with the creation date:

   ```
   \>type CTYST.DIR.
   ```
8. The first line of output will display the date (i.e. 4/2000). If you set the GSDRELEASEDATE to 20000415 then the CTYST.DIR date displayed will be 4/2000.
9. Copy the CTYST.DIR and the CTYST.DBF files into the `\gsdload\verMMYY\ctyst` directory and copy the CTYST.DIR into the `\gsdload\verMMYY` directory on the master build drive.

Sample batch file used to build ctyst.dir

```
set GSDRELEASEDATE=20000415
ctyst ctystate.txt CTYST.DIR
```

us.gsd

Relationships/dependencies upon other files

The us.gsd file is created by first building the individual state .gsd files, and then merging all the state .gsd files into the us.gsd file.

Build procedure for state .gsd files

If you have received the current month’s ZIP+4 files from the USPS then you will be able to skip steps 1 through 5. Due to the fact that we normally receive the current month’s ZIP+4 files around the middle of the month, it is too late to use them in the current month’s data build. Therefore, we use the previous month’s ZIP+4 files and the current month’s ZIP+4 Transaction file which is normally received by the 1st day of the current month along with the ZIP+4 Change file and City/State file.

NOTE: *???? always means 2-digit month and 2-digit year.*

1. After unloading the ZIP+4 files in the `\\lizard\D_lizard\uspsraw\zip4` directory, copy the ZIP+4 transaction file from where ever you unloaded it to the above directory.
2. Load DOS foxpro and run the break.prg program in the `\\lizard\D_lizard\uspsraw\zip4` directory. This program reads the z4trans.txt file and outputs 3-digit ZIP code .DBF files (e.g. ZIP code 80301 would be broken out of the z4trans.txt file into a 803.dbf file). These files will correspond to existing .ZIP ZIP+4 files that were unloaded to the `\\lizard\D_lizard\uspsraw\zip4` directory.
3. Edit the major.prg program in a text editor. Change the array listing of files to match the ZIP+4 .ZIP files in the `\\lizard\D_lizard\uspsraw\zip4` directory. Also, change the For loop to match the array size that was just changed.
4. In DOS foxpro, run major.prg. This program reads the 3-digit zip code .DBF files, created earlier and determines if the USPS records contained within are Append, Insert or Delete type records. It then makes the needed changes to those records in the corresponding ZIP+4 file.
5. Copy the updated 3-digit zip code files to the `\\thistle\thistle_g\zip4` directory.
6. Load RR (DOS Report Writer) from the `\\middleman\sdrive\gsdload\ver????\bats` directory and edit the USPS pass3 std batch file. Make changes for the new build (i.e. location of current version of gsdload directory, current USPS us.gsd file, GSDRELEASE-DATE, etc.).
7. In RR, output changed batch file to an ASCII file in the mkp3 directory under the bats directory that you ran RR from. When RR outputs this file it will contain a copy of the batch file for each state and territories of the US. There are 62 total when building the USPS us.gsd.

   Note: When merging any us.gsd file remember there has to be 62 state gsd files, if you have any less than this then you are missing some.
8. Open the batch file in a text editor and split it up into 4 batch files, each containing an equal number of states (i.e. 62 / 4). Name the file run1.bat through run4.bat.
9. Run each of these batch files on a separate build computer.
10. Before running any batch file, for any step in the build process do the following:

    On the D:\ of each build computer create a `ver????` (????→ current month and year) directory under the `d:\gsdload` directory.
11. Copy *.* from `\\middleman\sdrive\gsdload\ver????` (????→ current month and year) into the newly created `ver????` directory on the build computers. Make sure that the directory name on the build computers is the same as the directory on middleman’s S:\. The files copied will be the newest DIR files, built earlier, and all of the utilities needed to build USPS, GDT and TIGER data, as well as the needed libraries (i.e. Geostan, Spatial), and needed GSB files.
12. Create a new directory on the root of D:\, on the build computer, called `usps????`.

13. Copy one batch file from the `\\middleman\sdrive\gsdload\ver????\bats\mkp3` directory into the `usps????` directory on the local build computer.
14. From the command prompt run the batch file with the following command line parameters:

    `D:\usps????\>Run1.bat D: G:\zip4 S: S:`

    - The first command line parameter is the drive location of the newest gsdload directory.
    - The second command line parameter is the directory location of the ZIP+4 USPS raw files.
    - The third command line parameter is the drive location to move the Building DBF files to.
    - The fourth command line parameter is the drive location to move the state GSD files and the state Z4 zip files to.
15. Before running this batch file look over the batch file and make sure that all of the directories referenced in the batch file actually exist on the drives you are declaring in the command line arguments. Also, in the directory where the batch file is moving the state Z4 zip and GSD files to create a sub directory for all 62 state and territories as well as a state and other directories.
16. Run the batch file. This process will take a full workday to run.
17. Perform the same operations of the other for build computers used to build the USPS state gsd files.

Sample batch file used to build a single state .gsd file

This is an example of only one state. In the actually batch file you would have this procedure repeated 62 times, each for a different state, territory or military base. This batch file (w/62 repeated procedures) would be broken up into 4 parts to run on 4 build computers.

```
REM -----------------------------
REM State Processing pass3 for AZ
REM USPS First Step Run ALIAS after this
REM -----------------------------
REM First run the ctyst process to create a new ctyst.dir from usps
REM note the ZIP4 dir is on the USPS CD
REM -----------------------------
SET GSDRELEASEDATE=20000415
SET GSDLOGFILE=S:\GSDLOGS\USPS0400.DBF
cd \
cd \usps0400
mkdir AZ
cd AZ
%1\GSDLOAD\VER0400\PASS3 . AZ %2\ZIP4 %1\GSDLOAD\VER0400 /B
call GETSCC
%1\GSDLOAD\VER0400\PASS3 . AZ . %1\GSDLOAD\VER0400 /R /U
%1\GSDLOAD\VER0400\BUILDING /C . AZ %1\GSDLOAD\VER0400
move 04BLDNG.DBF %3\BUILD04
move *.gsd %4\usps0400\STATE
PKZIP  AZZ4 *.dbf
copy AZZ4.ZIP %4\usps0400\AZ
del *.cdx
del *.dbf
```

Merging state .gsd files into the us.gsd

NOTE: *MMYY means 2-digit month and 2-digit year.*

1. On the build computer used to merge the USPS State GSD files, create a directory on the root of D:\ called `USPSMMYY.`
2. Create two sub-directories called `build` and `state.`
3. Copy all Building DBF files into the build directory. The Building DBF files are located where the USPS State GSD batch files moved them to, normally `\\middleman\sdrive\buildMM` (MM means current 2-digit month).
4. Copy the 62 GSD files into the state directory, from where the USPS State GSD batch files moved the GSD files to. This is normally the `\\middleman\sdrive\uspsMMYY\state` directory.
5. Before performing the above step, copy the following state, territory and military base GSD files into the other directory under the `\\middleman\sdrive\uspsMMYY` directory. These GSD files will be used later for the merging of the TIGER and GDT US.GSD:

   AA.GSD, AE.GSD, AP.GSD, AS.GSD, FM.GSD, GU.GSD, MH.GSD, MP.GSD, PW.GSD, VI.GSD.
6. Copy the contents of the `\\middleman\sdrive\gsdload\verMMYY\usgsd` directory into the `USPSMMYY` directory on the D:\ of the local build computer.
7. Edit the mkem.bat file in a text editor. Change the GSDRELEASEDATE and uncomment the uniq and gsdmerge lines of the batch file. Change the uniq line so that it references the correct state directory (this is the state directory where you copy the state GSD files to).

   NOTE: The reason you copied the state GSD files is because the uniq batch file makes changes to the GSD files and you do not want to overwrite the originals in case you have to merge again.

   Change the gsdmerge line of the batch file so that it is referencing the correct state directory and so that the data type is USPS data and not GDT.
8. Edit the uniq.bat so that the location of the state Z4 zip files is correct; change this directory for every state in the uniq.bat file.
9. Copy the CTYST.DIR, MSAC.DIR and PARSE.DIR files from the current `\\middleman\sdrive\gsdload\verMMYY` directory to the `USPSMMYY` directory on the D:\ of the local build computer.
10. Run the mkem.bat batch file.
11. This process will output the USPS data only US.GSD file that is used in the process to build the street segment STATE and US GSD files.

parse.dir

The parse.dir file contains a series of tables which GeoStan uses for parsing. It contains the following information (most of which are defined in USPS publication 28):

- Generic spelling variations and abbreviations (e.g. "ENGR" for "ENGINEER")
- USPS street types and unit types (e.g. ST for street)
- USPS street type and unit type spelling variations (e.g. STR and STREET)

- USPS state abbreviations (e.g. MN for Minnesota)
- USPS state spelling variations (e.g. MINN, MINNESOTA, and MN)
- Common spelling variations in city names (e.g. ST for Saint)

Most of these tables are repeated within GeoStan or Qlib source in hard-coded arrays.

Typically, parse.dir is created once each year during CASS certification and is not modified again until the next CASS cycle.

Relationships/dependencies upon other files

This file is built using only USPS data, but is required to run GeoStan.

The parse.dir file is built using the information from a series of other files; see the explanation of the *input_files_path* parameter for loadabrv.exe in the following build procedure for a list of these files.

Build procedure

The program loadabrv.exe is used to build the parse.dir file, and is stored in PVCS under the project `\gsd\abbrev.`

The program loadabrv.exe has the following command-line syntax:

`loadabrv` *`input_files_path output_file`*

`input_files_path`Directory containing the input files. These files are stored in PVCS under the project `\gsd\abbrev\tables`. The input files are:

- abbrevat.txt
- states.txt
- suffixes.txt
- unittype.txt
- parse.dct
- lastline.dct
- states.tbl
- strtype.tbl

`output_file`For GeoStan, this name must be "parse.dir" because the file name is hard-coded into the GeoStan library and data build programs.

Many of the input files contain duplicate information, but the redundancy is required for the information to be implemented in the entire program. For example, to add a new street type abbreviation, you must modify suffixes.txt, parse.dct and strtype.tbl. To add a completely new street type, you must also modify the PostalStreetType array in Qlib.

The release/build date located in the parse.dir file header is set with the environment variable GSDRELEASEDATE.

The following is an example of how parse.dir is built using two DOS commands:

```
set GSDRELEASEDATE = 20000415
s:\gsd\abbrev\debug\loadabrv s:\gsd\abbrev\tables parse.dir
```

msac.dir

This file is used by only 2-3 customers, and has changed very little in the past few years.

Relationships/dependencies upon other files

This file is built using only USPS data, but is required to run GeoStan.

Build procedure

The application msabuild.exe is used to build the msac.dir file.

us.gsl

Relationships/dependencies upon other files

The ltravel.txt file that is used for input to the GSLBUILD.EXE utility is normally one month behind due to the fact that the ltravel.txt file comes on the USPS ZIP+4 CD's, which ship to us around the middle of the current month. Since the data files are due to testing as early as possible we normally build the US.GSL with the current Z4Change.txt file and the previous month's ltravel.txt file. This will affect CASS testing due to the fact that we are using the current USPS ZIP+4 data and a month old Line-of-Travel (ltravel.txt) data. The only way around this is to wait for the current USPS ZIP+4 data to arrive and build the US.GSL then.

Build procedure

The build procedure for us.gsl is the same for both TIGER and GDT data.

The drive used to build the US.GSL should have ~6GB of free space as the US.GSL build process takes up a lot of temporary space.

1. On a local drive create a directory called `Z4change`.
2. Unload the Z4Change.txt file from tape to the above directory. The Z4Change tape set should have been received at the beginning of the month with the City/ State and the ZIP+4 Transaction tapes sent from the USPS. The Z4Change.txt is over 1GB in size when unloaded from tape.
3. Copy the ltravel.zip file from the most current set of USPS ZIP+4 CDs (normally on CD#2 of the set) to the Z4change directory. The file is password protected, so when you unzip it you will need to enter the password "/H4R9KXF7A3". Unzip the ltravel.txt file
4. Copy the GSLBUILD.EXE utility from the most current gsdload version `\\middleman\sdrive\gsdload\verMMYY` to the `Z4change` directory.
5. Create a directory in the root of the local build drive called temp.
6. Set the TEMP system variable to the above directory.
   Set `TEMP=localdrive:\temp.` The local drive must be the same as the drive with the Z4change directory. If `D:\Z4change`, then temp must be `D:\temp.`
7. Set the GSDRELEASEDATE system variable to the current month (i.e. 20000415 for April of 2000): set `GSDRELEASEDATE=20000415.`
8. Run the GSLBUILD.EXE utility with the following command:

```
D:\z4change>gslbuild ltravel.txt z4change.txt US.GSL
```

The ltravel.txt and the z4change.txt files can be either in a text format or a DBF format. If the files are in a text format then GSLBUILD.EXE will first convert them to a DBF format file before continuing with processing.

Sample batch file used to build us.gsl

```
set GSDRELEASEDATE = 20000415
gslbuild ltravel.txt US.GSL
```

Data files built from TIGER data

us.gsd

Relationships/dependencies upon other files

The us.gsd file is created by first building the individual state .gsd files, and then merging all the state .gsd files into the us.gsd file.

TIGER data only covers the 50 US states as well as D.C. and Puerto Rico for a total of 52. The remaining 10 territories and military bases are comprised of only USPS data but are still merged into the final US.GSD since they contain needed USPS information.

Build procedure for state .gsd files

1. The first step to perform is to copy the USPS data only US.GSD from the `\\middleman\sdrive\uspsMMYY\us` directory to D:\ on the local computer. Normally we copy the USPS US.GSD into the following directory:

   `D:\uspsMMYY\us`

   NOTE: You can place this where you want it, but make sure you modify the batch files to reference this location. Refer to the example batch file after this section for more information.

2. Load RR (DOS Report Writer) from the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Retrieve the TGR Process Pass1 – ALL report. Edit this report for the current GSDRELEASEDATE, current USPSMMYY directories, current GSDLOAD version, etc. An example of this batch file follows this section.

3. In RR, output changed batch file to an ASCII file in the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Name it TGR98.BAT. When RR outputs this file it will contain a copy of the example batch file for each state, territory and military base. There are 52 total when building a TIGER street segment data US.GSD. The extra 2 states are for Puerto Rico and the District of Colombia.

   NOTE: When merging any us.gsd file remember there has to be 62 state .gsd files, not 52. If you have any less than this then you are missing some.

4. Go to the `\\middlman\sdrive\bats\TGR` directory and del *.* from the directory.

5. Copy the TGR98.BAT file to the `\\middleman\sdrive\bats\TGR` directory.

6. Use the FILSPLIT.EXE utility to split the TGR98.BAT file into 62 state batch files (this utility is stored in the `C:\utils\system` directory of each build computer and is set in the environment PATH variable).

   Command line: `localdrive:\>filsplit TGR98.bat`

   FILSPLIT.EXE reads the file in and splits the file based on the individual states it encounters in the file. The output will be 62 batch files named as follows:

   TGR_AZ.BAT (for the state of Arizona)

7. On the local build computer, create a directory called `tgrgsd` in the root of D:\.
8. Determine how many build computers are available for processing. You should always try to have 8 available. This way you can split up the processing load evenly across multiple build computers
9. Copy the pre-determined number of the split TIGER batch files from the `\\middlman\sdrive\bats\TGR` directory to the newly created `D:\tgrgsd` directory.

   There will be 62 total TIGER batch files in the `\\middlman\sdrive\bats\tgr` directory. You will only want to use the 52 state batch files which there is TIGER street segment data for. Do not use the batch files for the 10 territories and military bases:

   AA, AE, AP, AS, FM, GU, MH, MP, PW, VI.

10. Create and edit a batch file that will call each of the individual TIGER state batch files and pass those batch files the required command line arguments. An example of this batch file is below:

    Example of a TIGER state GSD MKEM.BAT file

```
call TGR_CO x:\tgr98 d: d:
call TGR_CT x:\tgr98 d: d:
call TGR_DC x:\tgr98 d: d:
call TGR_DE x:\tgr98 d: d:
call TGR_FL x:\tgr98 d: d:
call TGR_GA x:\tgr98 d: d:
call TGR_HI x:\tgr98 d: d:
call TGR_IA x:\tgr98 d: d:
call TGR_ID x:\tgr98 d: d:
```

11. The command line arguments for the TIGER state batch files are listed below:

    `X:\tgr98`→location of the TIGER raw data files (%1 variable in the TIGER sample batch file).

    `D:`→local location of current `GSDLOAD\VERMMYY` directory (%2 variable in the TIGER sample batch file).

    `D:`→local location of the USPS US.GSD file (%3 variable in the TIGER sample batch file).

12. Look through one of the TIGER state batch files to make certain that dates and directory names are correct and the directories referenced in the file actually exist and have the files needed contained in them. The important directories for this process are listed below:

    - Correct GSDRELEASEDATE.
    - Location of the TIGER raw data files.
    - Location of the USPS US.GSD file.

- Location of the GSD log file.
- Location of the current build utilities (verMMYY).
- Location for the placement of the TIGER centroid files.

13. Make sure that the 5Zip.dbf file is contained in the TIGER centroid directory. This is normally in the `\\middleman\sdrive\tgrcent` directory.
14. After making sure that all of the needed utilities and data files are located where they should be and that all of the needed directories are made, run the MKEM.BAT batch file.

    NOTE: The first state you start will give you a message that the TIGER log file does not exist and should it create it. Say yes to this message and then restart the entire process, as the state that was being worked on when the message came up will have failed. Make sure the directory that was created for the state is deleted before restarting.
15. Perform the same operation on all of the other build computers so that all of the 52 TIGER state batch files get processed.
16. If you are running 6 or 7 files per build computer, the entire process should take a couple of days.
17. The output from this process is 52 primary state.gsd files, 52 supplemental state.gsd files and 52 state.zip back-up files. The state zip files contain the state primary and supplemental GSD files, logs and intermediate DBF files.

Sample batch file used to build a single state .gsd file

```
SET GSDRELEASEDATE=20000415
SET GSDLOGFILE=S:\GSDLOGS\TGR0400.DBF
cd \
cd \TGRGSD
MKDIR CA
cd CA
IF ERRORLEVEL 1 GOTO ERRTRAP
mkdir TIG
IF ERRORLEVEL 1 GOTO ERRTRAP
cd TIG
IF ERRORLEVEL 1 GOTO ERRTRAP
copy %1\TIGER\06_CA\*.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
for %%f in (*.ZIP) do pkunzip %%f *.RT1 *.RT2 *.RT3 *.RT4 *.RT5 *.RT6 *.RTa
*.RTi
IF ERRORLEVEL 1 GOTO ERRTRAP
rename TGR?????.RT? TGR?????.BW?
IF ERRORLEVEL 1 GOTO ERRTRAP
del *.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
cd ..
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS1 TIG .  %3\USPS0400\US;%2\GSDLOAD\VER0400 CA /95
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS2 . CA %2\GSDLOAD\VER0400
IF ERRORLEVEL 1 GOTO ERRTRAP
del 061.DBF
IF ERRORLEVEL 1 GOTO ERRTRAP
copy s:\usps0400\CA\CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
pkunzip CAZ4 06Z4.DBF
```

```
IF ERRORLEVEL 1 GOTO ERRTRAP
del CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS3 . 06 . %2\GSDLOAD\VER0400 /C
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS3 . 06 . %2\GSDLOAD\VER0400 /T
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\ZIPCENT . S:\TGRCENT CA
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS4 . CA S:\TGRCENT
IF ERRORLEVEL 1 GOTO ERRTRAP
REM START CLEAN UP
del *.CDX
del ??Z4.dbf
pkzip TIG_CA *.*
IF ERRORLEVEL 1 GOTO ERRTRAP
del ??P2.dbf
del ??_SHPE
del 06CENT.DBF
GOTO EXIT
:ERRTRAP
ECHO CA has had a failure > ERROR.TXT
:EXIT
cd \TGRGSD
REM ---------------------------------------------------
```

Merging state .gsd files into the us.gsd

NOTE: MMYY stands for the current 2-digit month and 2-digit year.

1. On the computers that ran the TIGER state batch files create a batch file called MVEM.BAT in the `D:\tgrgsd` directory.

   Example of mvem.bat batch file

   ```
   for /r %%f in (??.gsd) do move %%f s:\tgr0400\state
   for /r %%f in (??t.gsd) do move %%f s:\tgr0400\sup
   for /r %%f in (*.zip) do move %%f z:\tgr04bck
   ```

   This batch file will recursively search through all of the sub-directories of the `D:\tgrgsd` directory for state primary or supplemental state GSD files as well as any state zip files. It will then move the primary GSD files to a single location, the supplemental GSD files to a separate location, and the State zip files to a back-up directory. The previous example is for the April 2000 build. We normally move all of the state GSD files to `\\middleman\sdrive` so that they are accessible from any of the other build computers. The zip back-up files are normally stored on `\\lizard\zdrive`.

2. Run MVEM.BAT on one build computer at a time

3. Count the number of primary state GSD files in the `\\middleman\sdrive\tgrMMYY\state` directory; there should be 52. If there are less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state’s batch file failed. If the latter is the case go to the computer that ran the specific state and in that state’s build directory, under `D:\tgrgsd`, look at the logs to determine how it failed.

4. Count the number of supplemental state GSD files in the `\\middleman\sdrive\tgrMMYY\sup` directory; there should be 52. If there are less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case, go to the computer that ran the specific state and in that state's build directory, under `D:\tgrgsd`, look at the logs to determine how it failed.
5. Copy all the GSD files from the `\\middleman\sdrive\uspsMMYY\other` directory (where MMYY is the same as `\\middleman\sdrive\tgrMMYY`) into the `\\middleman\sdrive\tgrMMYY\state` directory. Now the total primary state GSD files in the state directory should be the needed 62. If there are less check everything as in step 4 above and check to see that the `\\middleman\sdrive\uspsMMYY\other` directory has the required 10 USPS data only GSD files:

   AA.GSD, AE.GSD, AP.GSD, AS.GSD, FM.GSD, GU.GSD, MH.GSD, MP.GSD, PW.GSD, VI.GSD
6. In the root of the local D: drive, create a new directory called `tgrMMYY.`
7. Also in the root of the local D: drive, create 2 other directories called `tgrext` and `tgralts.`
8. Create 3 sub-directories under the `tgrMMYY` directory called `state`, `sup` and `build.`
9. Copy ctystate.txt from the `\\middleman\sdrive\gsdload\verMMYY\ctyst` directory into the `D:\tgrMMYY` directory.
10. Copy the current CTYST.DIR, MSAC.DIR and PARSE.DIR files from the `\\middleman\sdrive\gsdload\verMMYY` directory into the `D:\tgrMMYY` directory.
11. In the `D:\tgrMMYY\state` directory, copy the 62 state GSD files from the `\\middleman\sdrive\tgrMMYY\state` directory.
12. In the `D:\tgrMMYY\build` directory, copy the 62 building DBF files from the `\\middleman\sdrive\buildMM` directory. These files are named in the following format: ??bldng.dbf, where ?? stands for the corresponding state, territory or military base's FIPS code. There should never be less or more than 62 of these building DBF files.
13. In the `D:\tgrMMYY\sup` directory, copy the 52 supplemental GSD files from the `\\middleman\sdrive\tgrMMYY\sup` directory.
14. In the `D:\tgrext` directory, copy the contents of the `\\middleman\sdrive\gsdload\verMMYY` directory.
15. In the `D:\tgrMMYY` directory, edit the MKEM.BAT batch file.
    - Uncomment all lines.
    - Change GSDRELEASEDATE to the correct date.
    - Make sure directories and files exist.
    - Make sure the data type specified for the GSDMERGE.EXE utility is correct.
    - Make sure that the locations of the state primary and supplemental GSD and building DBF file locations are correct for the GSDMERGE.EXE and BUILDING.EXE utilities.
16. Also in the `D:\tgrMMYY` directory, edit the UNIQ.BAT batch file.

- Make sure that the location of the state Z4 zip files is correct for the current build.
- You will have to change the directory location for all 62 state, territory, and military base locations.

17. In the `D:\tgrext` directory, edit the RUNALIAS.BAT batch file.
    - Make sure all directories and files exist.
    - Make sure the location of the TIGER state zip backup files is correct for the current build.
    - Make sure that the locations of US.GSD and UST.GSD are correct.
    - NOTE: US.GSD and UST.GSD will be created prior to the call to the runalias.bat file.
18. Recheck the existence and locations of all directories and files, then run MKEM.BAT.

Example of mkem.bat

```
SET GSDRELEASEDATE=20000415
call uniq .\state\
gsdmerge .\state US.GSD TIGER /nad83
building .\build US.GSD US.GSD .
alias ctystate.txt US.GSD US.GSD .
gsdmerge .\sup UST.GSD TIGER /s /nad83
cd \tgrext
call runalias
```

Example of uniq.bat

This is an example of only one state out of the 62 state, territory and military bases. This same example is repeated, in the UNIQ.BAT file, for each of the 62 locations.

```
pkunzip S:\USPS0400\AZ\AZZ4
uniquez4 04Z4.DBF CTYST.DIR PARSE.DIR %1AZ.GSD
del 04Z4.DBF
del 04ALT.DBF
```

Example of runalias.bat

```
cd \tgralts
for %%f in (z:\tgr04bck\*.zip) do pkunzip %%f ??A.DBF
cd \tgrext
spalias d:\tgralts .;d:\tgr0400 /fix
spalias d:\tgralts d:\tgr0400\us.gsd us.gsi .;d:\tgr0400
copy us.gsi us_P.gsi
spalias d:\tgralts d:\tgr0400\ust.gsd us.gsi .;d:\tgr0400
copy us.gsi us_PS.gsi
stintbld d:\tgr0400\us.gsd d:\tgr0400\ust.gsd d:\tgr0400\;.
us.gsi
```

us_p.gsi

Relationships/dependencies upon other files

The us.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd as an input file, and the spalias utility.

us_ps.gsi

Relationships/dependencies upon other files

The ust.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using ust.gsd as an input file, and the spalias utility.

us.gsi

Relationships/dependencies upon other files

The us.gsd and ust.gsd files for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd and ust.gsd as input files, and the stintbld utility.

ust.gsd

Relationships/dependencies upon other files

This file is composed of the 52 supplemental state .gsd files, which are then merged into ust.gsd by the mkem.bat file.

Build procedure

This file is built by merging the 52 state supplemental .gsd files using the gsdmerge utility with the \s flag. This file is built as part of the merge process for us.gsd; the file does not need to be built separately.

us.z9

Relationships/dependencies upon other files

The only difference between a US.Z9 for the different data types is the centroid files used as input to this process. For example, if building the TIGER US.Z9 then the input for this process would be the TIGER centroid files produced during the TIGER state GSD file build process.

Build procedure

??? stands for one of the street segment data types. GDT→GDT, TGR→ TIGER, NAV→Navigation Technologies, GEO→Geosystems, etc.

Steps 6 and 7 are performed because these are known data problems. More data issues could come up in the future but right now this is all we know about with the Centroid files.

1. On the local build computer's D: drive, create a directory called `???cent` for the type of US.Z9 that is being built.
2. Create a batch file called MKCENTS.BAT. See the example following this section. This batch file sets the GSDRELEASEDATE environment variable, then runs the ZIP4LOAD.EXE utility from the current `GSDLOAD` directory (make sure the directory name is correct for the current build) to build the US.Z9 file from the current build's centroid files (make sure that the centroid file location is correct for the current build). The ZIPINDEX.EXE utility file is then used to index the newly created US.Z9.
3. In the data types centroid directory, normally `\\middlman\sdrive\???cent`, create a sub-directory called `7zip.`
4. In the `\\middlman\sdrive\???cent` directory, move all of the 7zip??.dbf files into the `7zip` directory (?? stands for the corresponding state's FIPS code).

   NOTE: We do this because we do not want to build the US.Z9 with the 7-digit centroid files, only the 5- and 9-digit centroid files. We output the 7-digit centroid files so that we have the option.
5. Delete *.idx (index file) from the `\\middlman\sdrive\???cent` directory.
6. Open the 9zip26.dbf file (9-digit centroid file for MI).
   - Index the file based on the ZIP column.
   - Search for ZIP code 49950.
   - If the BlockGroup column contains a census tract of all 0s change it to 26083960300 for all census tracts that need it.
   - The format of the Census tract ID is:

     SSCCCTTTT.TT

     Where SS=State Code

     CCC=County Code

     TTTT.TT=Census Tract
7. Open the 9zip02.dbf file (9=digit centroid file for AK)
   - Index the file based on the ZIP column
   - Search for ZIP code 99761
     - Check to make sure that the CCC code is 185 and not 050
   - Search for ZIP code 99636
     - Check to make sure that the CCC code is 070 and not 050
   - Search for ZIP code 99780
     - Check to make sure that the CCC code is 261 and not 240
8. In the `D:\???cent` directory, run the MKCENTS.BAT file.

Sample batch file used for build

The following is an example of the mkcents.bat file:

```
SET GSDRELEASEDATE=20000415
d:\gsdload\ver0400\zip4load US.Z9 s:\tgrcent
d:\gsdload\ver0400\zipindex US.Z9
```

Data files built from GDT data

us.gsd

Relationships/dependencies upon other files

The us.gsd file is created by first building the individual state .gsd files, and then merging all the state .gsd files into the us.gsd file.

GDT data only covers the 50 US states as well as D.C. and Puerto Rico. The remaining 10 territories and military bases are comprised of only USPS data but are still merged into the final US.GSD since they contain needed USPS information.

Creating GDT raw files from CD

This procedure is used to create raw GDT files for the state .gsd file build.

1. Copy the contents of the CD to the hard drive (preserve the directory structure):

   ```
   xcopy e:\ d:\ /s
   ```

   e:\ is the CD-ROM drive

   d:\ is the destination hard drive

   Make sure that you copy the full states.

2. In the newly created GDT directory, do

   ```
   for /r %f in (D?????.tx?)
   do rename %f D?????s.tx?
   for /r %f in (D?????.tp?)
   do rename %f D?????P.tx?
   ```

3. Edit mkem.bat
   - Put a copy of mkem.bat in the USA directory
   - The format for each state in mkem.bat is the following:

     ```
     cd ??
     pkzip -r gdt?? *.*
     cd ..
     ```

     ?? is the state directory

4. From the USA directory, run mkem.bat.
5. Check the size of the old raw files vs. the zipped files.
6. In the raw directory, do:
   - In the USA directory,

     ```
     for /r %f in (*.zip) do copy %f destination/store
     (i.e. g:\gdtioraw)
     ```

- copy GDT zip files to storage directory

7. Remove the state directories
   - From within the USA directory, do

     ```
     for /D %f in (??) do rd /s /q %f
     ```

     ?? is the 2-digit state directory name

Build procedure for state .gsd files

1. The first step to perform is to copy the USPS data only US.GSD from the `\\middleman\sdrive\uspsMMYY\us` directory to the D:\ on the local computer. Normally we copy the USPS US.GSD into the following directory:

   `D:\uspsMMYY\us`

   NOTE: You can place this were you want it, but make sure you modify the batch files to reference this location. Look at the example batch file at the end of this document for more information.
2. Load RR (DOS Report Writer) from the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Retrieve the GDT Process Pass1 – ALL report. Edit this report for the current GSDRELEASEDATE, current `USPSMMYY` directories, current GSDLOAD version, etc. An example of this batch file follows this section.
3. In RR, output changed batch file to an ASCII file in the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Name it GDTALL.BAT. When RR outputs this file it will contain a copy of the example batch file for each state, territory and military base. There are 52 total when building a GDT street segment data US.GSD. The extra 2 states are for Puerto Rico and the District of Colombia.

NOTE: when merging any us.gsd file remember there has to be 62 state gsd files, not 52. If you have any less than this then you are missing some.

4. Go to the `\\middlman\sdrive\bats\gdt` directory and del *.* from the directory.
5. Copy the GDTALL.BAT file to the `\\middleman\sdrive\bats\gdt` directory.
6. Use the FILSPLIT.EXE utility to split the GDTALL.BAT file into 62 state batch files. This utility is stored in the `C:\utils\system` directory of each build computer and is set in the environment PATH variable.

   Command line: `localdrive:\>filsplit gdtall.bat`

   FILSPLIT.EXE reads the file in and splits the file based on the individual states it encounters in the file. The output will be 62 batch files named as follows:

   GDT_AZ.BAT (for the state of Arizona).
7. On the local build computer, create a directory called `gdtgsd` in the root of D:\.
8. Determine how many build computers are available for processing. You should always try to have 8 available. This way you can split up the processing load evenly across multiple build computers.
9. Copy the pre-determined number of the split GDT batch files from the `\\middlman\sdrive\bats\gdt` directory to the newly created `D:\gdtgsd` directory.

There will be 62 total GDT batch files in the `\\middlman\sdrive\bats\gdt` directory. You will only want to use the 52 state batch files which there is GDT street segment data for. Do not use the batch files for the 10 territories and military bases:

AA, AE, AP, AS, FM, GU, MH, MP, PW, VI.

10. Create and edit a batch file that will call each of the individual GDT state batch files and pass those batch files the required command line arguments. An example of this batch file is below:

    Example of a GDT state GSD MKEM.BAT file:

    ```
    call GDT_CO x:\gdt10raw d: d:
    call GDT_CT x:\gdt10raw d: d:
    call GDT_DC x:\gdt10raw d: d:
    call GDT_DE x:\gdt10raw d: d:
    call GDT_FL x:\gdt10raw d: d:
    call GDT_GA x:\gdt10raw d: d:
    call GDT_HI x:\gdt10raw d: d:
    call GDT_IA x:\gdt10raw d: d:
    call GDT_ID x:\gdt10raw d: d:
    ```

11. The command-line arguments for the GDT state batch files are listed below:
    - X:\gdt10raw→location of the GDT raw data files (%1 variable in the GDT sample batch file). See “Creating GDT raw files from CD” on page 24 for more information on these files.
    - D:→local location of current `GSDLOAD\VERMMYY` directory (%2 variable in the GDT sample batch file).
    - D:→local location of the USPS US.GSD file (%3 variable in the GDT sample batch file).
12. Look through one of the GDT state batch files to make certain that dates and directory names are correct and the directories referenced in the file actually exist and have the files needed contained in them. The important directories for this process are listed below:
    - Correct GSDRELEASEDATE.
    - Location of the GDT raw data files.
    - Location of the USPS US.GSD file.
    - Location of the GSD log file.
    - Location of the current build utilities (verMMYY).
    - Location for the placement of the GDT centroid files.
13. Make sure that the 5Zip.dbf file is contained in the GDT centroid directory. It is normally in the `\\middleman\sdrive\gdtcent` directory.
14. After making sure that all of the needed utilities and data files are located where they should be and that all of the needed directories are made, run MKEM.BAT batch file.

    NOTE: The first state you start will give you a message that the GDT log file does not exist and should it create it. Say yes to this message and then restart the entire process, as the state that was being worked on when the message came up, will have failed. Make sure the directory that was created for the state is deleted before restarting.
15. Perform the same operation on all of the other build computers so that all of the 52 GDT state batch files get processed.

16. If you are running 6 or 7 files per build computer, the entire process should take a couple of days.
17. The output from this process is 52 primary State.GSD files, 52 supplemental State.GSD files and 52 State .zip back-up files. The state zip files contain the state primary and supplemental GSD files, logs and intermediate DBF files.

Sample batch file used to build a single state .gsd file

```
REM State Processing ALL PASSES
REM %1 = GDT RAW DATA DRIVE
REM D:    or    S:\GDTRAW
REM %3 = DRIVE for US USPS file
REM %2 = LOCAL DRIVE
REM ----------------------------------------
SET GSDRELEASEDATE=20000415
SET GSDLOGFILE=S:\GSDLOGS\GDT0400.DBF
cd \
cd \GDTGSD
MKDIR CA
cd CA
IF ERRORLEVEL 1 GOTO ERRTRAP
MKDIR GDT
IF ERRORLEVEL 1 GOTO ERRTRAP
CD GDT
IF ERRORLEVEL 1 GOTO ERRTRAP
copy %1\GDT06.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
pkunzip GDT06
IF ERRORLEVEL 1 GOTO ERRTRAP
del GDT06.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
CD ..
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS1 GDT . %3\USPS0400\US;%2\GSDLOAD\VER0400 CA /G
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS2 . CA %2\GSDLOAD\VER0400
IF ERRORLEVEL 1 GOTO ERRTRAP
del 061.DBF
IF ERRORLEVEL 1 GOTO ERRTRAP
copy s:\usps0400\CA\CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
pkunzip CAZ4 06Z4.DBF
IF ERRORLEVEL 1 GOTO ERRTRAP
del CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS3 . 06 . %2\GSDLOAD\VER0400 /N /C
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS3 . 06 . %2\GSDLOAD\VER0400 /N /T
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\ZIPCENT . S:\GDTCENT CA
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0400\PASS4 . CA S:\GDTCENT
IF ERRORLEVEL 1 GOTO ERRTRAP
REM START CLEAN UP
del *.CDX
del ??Z4.dbf
pkzip GDT_CA *.*
IF ERRORLEVEL 1 GOTO ERRTRAP
```

```
del ??P2.dbf
del ??_SHPE
del 06CENT.DBF
del 06A.dbf
GOTO EXIT
:ERRTRAP
ECHO CA has had a failure > ERROR.TXT
:EXIT
cd \GDTGSD
REM ---------------------------------------------
```

Merging state .gsd files into the us.gsd

NOTE: *MMYY stands for the current 2-digit month and 2-digit year.*

1. On the computers that ran the GDT state batch files create a batch file called MVEM.BAT in the `D:\gdtgsd` directory.

   Example of mvem.bat batch file:

   ```
   for /r %%f in (??.gsd) do move %%f s:\gdt0400\state
   for /r %%f in (??t.gsd) do move %%f s:\gdt0400\sup
   for /r %%f in (*.zip) do move %%f z:\gdt04bck
   ```

   This batch file will recursively search through all of the sub-directories of the `D:\gdtgsd` directory for and state primary or supplemental state GSD files as well as any state zip files. It will then move the primary GSD files to a single location, the supplemental GSD files to a separate location and the State zip files to a back-up directory. The example above is for the April, 2000 build. We normally move all of the state GSD files to `\\middleman\sdrive` so that they are accessible from any of the other build computers. The zip back-up files are normally stored on `\\lizard\zdrive`.
2. Run the MVEM.BAT on one build computer at a time
3. Count the number of primary state GSD files in the `\\middleman\sdrive\gdtMMYY\state` directory; there should be 52. If there is less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case go to the computer that ran the specific state and in that state's build directory, under `D:\gdtgsd`, look at the logs to determine how it failed.
4. Count the number of supplemental state GSD files in the `\\middleman\sdrive\gdtMMYY\sup` directory; there should be 52. If there is less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case go to the computer that ran the specific state and in that states build directory, under `D:\gdtgsd`, look at the logs to determine how it failed.
5. Copy all the GSD files from the `\\middleman\sdrive\uspsMMYY\other` directory (where MMYY is the same as `\\middleman\sdrive\gdtMMYY`) into the `\\middleman\sdrive\gdtMMYY\state` directory. Now the total primary state GSD files in the state directory should be the needed 62. If there are less check everything as in step 4 above and check to see that the `\\middleman\sdrive\uspsMMYY\other` directory has the required 10 USPS data only GSD files:

AA.GSD, AE.GSD, AP.GSD, AS.GSD, FM.GSD, GU.GSD, MH.GSD, MP.GSD, PW.GSD, VI.GSD

6. In the root of the local D: drive, create a new directory called `gdtMMYY.`
7. Also in the root of the local D: drive, create 2 other directories called `gdtext` and `gdtalts.`
8. Create 3 sub-directories, under the `tgrMMYY` directory, called `state`, `sup` and `build.`
9. Copy ctystate.txt from the `\\middleman\sdrive\gsdload\verMMYY\ctyst` directory into the `D:\gdtMMYY` directory.
10. Copy the current CTYST.DIR, MSAC.DIR and PARSE.DIR files from the `\\middleman\sdrive\gsdload\verMMYY` directory into the `D:\gdtMMYY` directory.
11. In the `D:\gdtMMYY\state` directory, copy the 62 state GSD files from the `\\middleman\sdrive\gdtMMYY\state` directory.
12. In the `D:\gdtMMYY\build` directory, copy the 62 building DBF files from `\\middleman\sdrive\buildMM` directory. These files are named in the following format:

    ??bldng.dbf, where ?? stands for the corresponding state, territory or military bases FIPS code.

    There should never be less or more than 62 of these building DBF files.
13. In the `D:\gdtMMYY\sup` directory, copy the 52 supplemental GSD files from the `\\middleman\sdrive\gdtMMYY\sup` directory.
14. In the `D:\gdtext` directory, copy the contents of the `\\middleman\sdrive\gsdload\verMMYY` directory.
15. In the `D:\gdtMMYY` directory, edit the MKEM.BAT batch file.
    - Uncomment all lines.
    - Change GSDRELEASEDATE to the correct date.
    - Make sure directories and files exist.
    - Make sure the data type specified for the GSDMERGE.EXE utility is correct.
    - Make sure that the locations of the state primary and supplemental GSD and building DBF file locations are correct for the GSDMERGE.EXE and BUILDING.EXE utilities.
16. Also in the `D:\gdtMMYY` directory, edit the UNIQ.BAT batch file.
    - Make sure that the location of the state Z4 zip files is correct for the current build.
    - You will have to change the directory location for all 62 state, territory and military base locations.
17. In the `D:\gdtext` directory, edit the RUNALIAS.BAT batch file.
    - Make sure all directories and files exist.
    - Make sure the location of the GDT state zip backup files is correct for the current build.
    - Make sure that the location of the US.GSD and UST.GSD are correct.

    NOTE: US.GSD and UST.GSD will be created prior to the call to the runalias.bat file.

18. Recheck the existence and locations of all directories and files, then run MKEM.BAT.

Example of mkem.bat

```
SET GSDRELEASEDATE=20000415
call uniq .\state\
gsdmerge .\state US.GSD GDT /nad83
building .\build US.GSD US.GSD .
alias ctystate.txt US.GSD US.GSD .
gsdmerge .\sup UST.GSD GDT /s /nad83
cd \gdtext
call runalias
```

Example of uniq.bat

This is an example of only one state out of the 62 state, territory and military bases. This same example is repeated, in the UNIQ.BAT file, for each of the 62 locations.

```
pkunzip S:\USPS0400\AZ\AZZ4
uniquez4 04Z4.DBF CTYST.DIR PARSE.DIR %1AZ.GSD
del 04Z4.DBF
del 04ALT.DBF
```

Example of runalias.bat

This batch file is called after completion of the US.GSD and UST.GSD merge. The output files from this process are US_P.GSI, US_PS.GSI and US.GSI

```
cd \gdtalts
for %%f in (z:\gdt04bck\*.zip) do pkunzip %%f ??A.DBF
cd \gdtext
spalias d:\gdtalts .;d:\gdt0400 /fix
spalias d:\gdtalts d:\gdt0400\us.gsd us.gsi .;d:\gdt0400
copy us.gsi us_P.gsi
spalias d:\gdtalts d:\gdt0400\ust.gsd us.gsi .;d:\gdt0400
copy us.gsi us_PS.gsi
stintbld d:\gdt0400\us.gsd d:\gdt0400\ust.gsd d:\gdt0400\;.
us.gsi
```

us_p.gsi

Relationships/dependencies upon other files

The us.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd as an input file, and the spalias utility.

us_ps.gsi

Relationships/dependencies upon other files

The ust.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using ust.gsd as an input file, and the spalias utility.

us.gsi

Relationships/dependencies upon other files

The us.gsd and ust.gsd files for this specific data type need to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd and ust.gsd as input files, and the stintbld utility.

ust.gsd

Relationships/dependencies upon other files

This file is composed of the 52 supplemental state .gsd files, which are then merged into ust.gsd by the mkem.bat file.

Build procedure

This file is built by merging the 52 state supplemental .gsd files using the gsdmerge utility with the **\s** flag. This file is built as part of the merge process for us.gsd; the file does not need to be built separately.

us.z9

Relationships/dependencies upon other files

The only difference between a US.Z9 for the different data types is the centroid files used as input to this process. For example, if building the GDT US.Z9 then the input for this process would be the GDT centroid files produced during the GDT state GSD file build process.

Build procedure

??? stands for one of the street segment data types. GDT→GDT, TGR→ TIGER, NAV→Navigation Technologies, GEO→Geosystems, etc.

Steps 6 and 7 are performed because these are known data problems. More data issues could come up in the future but right now this is all we know about with the Centroid files.

1. On the local build computer’s D: drive, create a directory called `???cent` for the type of US.Z9 that is being built.
2. Create a batch file called MKCENTS.BAT. See example following this section. This batch file sets the GSDRELEASEDATE environment variable, then runs the ZIP4LOAD.EXE utility from the current `GSDLOAD` directory (make sure the directory name is correct for the current build) to build the US.Z9 file

from the current build's centroid files (make sure that the centroid file location is correct for the current build). The ZIPINDEX.EXE utility file is then used to index the newly created US.Z9.

3. In the data types centroid directory, normally `\\middlman\sdrive\???cent`, create a sub-directory called `7zip.`
4. In the `\\middlman\sdrive\???cent` directory, move all of the 7zip??.dbf files into the `7zip` directory (?? stands for the corresponding state's FIPS code).

   NOTE: We do this because we do not want to build the US.Z9 with the 7-digit centroid files, only the 5- and 9-digit centroid files. We output the 7-digit centroid files so that we have the option.
5. Delete *.idx (index file) from the `\\middlman\sdrive\???cent` directory.
6. Open the 9zip26.dbf file (9-digit centroid file for MI).
   - Index the file based on the ZIP column.
   - Search for ZIP code 49950.
   - If the BlockGroup column contains a census tract of all 0's change it to 26083960300 for all census tracts that need it.
   - The format of the Census tract ID is:

     SSCCCTTTT.TT

     Where SS=State Code

     CCC=County Code

     TTTT.TT=Census Tract
7. Open the 9zip02.dbf file (9=digit centroid file for AK)
   - Index the file based on the ZIP column
   - Search for ZIP code 99761
     - — Check to make sure that the CCC code is 185 and not 050
   - Search for ZIP code 99636
     - — Check to make sure that the CCC code is 070 and not 050
   - Search for ZIP code 99780
     - — Check to make sure that the CCC code is 261 and not 240
8. In the `D:\???cent` directory, run the MKCENTS.BAT file.

Sample batch file used for build

The following is an example of a mkcents.bat file:

```
SET GSDRELEASEDATE=20000415
d:\gsdload\ver0400\zip4load US.Z9 s:\gdtcent
d:\gsdload\ver0400\zipindex US.Z9
```

us.epi

Relationships/dependencies upon other files

This file is built using only GDT data, and no other data type.

The GDT US.GSD, US.Z9 and CTYST.DIR, PARSE.DIR, and MSAC.DIR files need to be created prior to doing this process.

Build procedure

1. On the local D: drive, create a directory called `epiMMYY.`
2. Copy the contents of `\\middleman\sdrive\gsdload\verMMYY\indexbuild` to the `D:\epiMMYY` directory.
3. Copy the GDT US.GSD, US.Z9 and CTYST.DIR, PARSE.DIR and MSAC.DIR files to the `D:\epiMMYY` directory.
4. Run the INDEXBUILD.EXE utility with the following command line arguments:

   ```
   D:\epiMMYY>indexbuild .\ .\ US.EPI
   ```

   - The first “.\” is for the location of the Geostn32.dll.
   - The second “.\” is for the location of the required Geostan data files (step3).

Data files built from GEOSYSTEMS data

us.gsd

Relationships/dependencies upon other files

The us.gsd file is created by first building the individual state .gsd files, and then merging all the state .gsd files into the us.gsd file. The build procedure is almost exactly the same as for TIGER. The main thing is dir names are different.

Build procedure for state .gsd files

GEOSYSTEMS data only covers the 50 US states as well as D.C. and Puerto Rico for a total of 52. The remaining 10 territories and military bases are comprised of only USPS data but are still merged into the final US.GSD since they contain needed USPS information.

1. The first step to perform is to copy the USPS data only US.GSD from the \\middleman\sdrive`\uspsMMYY\us` directory to D:\ on the local computer. Normally we copy the USPS US.GSD into the following directory:

   ```
   D:\uspsMMYY\us
   ```

   NOTE: You can place this where you want it, but make sure you modify the batch files to reference this location. Refer to the example batch file after this section for more information.

2. Load RR (DOS Report Writer) from the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Retrieve the GEOSYS Process Pass1 – ALL report. Edit this report for the current GSDRELEASEDATE, current `USPSMMYY` directories, current GSDLOAD version, etc. An example of this batch file follows this section.
3. In RR, output changed batch file to an ASCII file in the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Name it GEOALL.BAT. When RR outputs this file it will contain a copy of the exam-

ple batch file for each state, territory and military base. There are 52 total when building a GEOSYSTEMS street segment data US.GSD. The extra 2 states are for Puerto Rico and the District of Colombia.

NOTE: When merging any us.gsd file remember there has to be 62 state .gsd files, not 52. If you have any less than this then you are missing some.

4. Go to the `\\middlman\sdrive\bats\GEO` directory and del *.* from the directory.
5. Copy the GEOALL.BAT file to the `\\middleman\sdrive\bats\GEO` directory.
6. Use the FILSPLIT.EXE utility to split the GEOALL.BAT file into 62 state batch files (this utility is stored in the `C:\utils\system` directory of each build computer and is set in the environment PATH variable).

   Command line: `localdrive:\>filsplit GEOALL.bat`

   FILSPLIT.EXE reads the file in and splits the file based on the individual states it encounters in the file. The output will be 62 batch files named as follows:

   GEO_AZ.BAT (for the state of Arizona)
7. On the local build computer, create a directory called `GEOgsd` in the root of D:\.
8. Determine how many build computers are available for processing. You should always try to have 8 available. This way you can split up the processing load evenly across multiple build computers
9. Copy the pre-determined number of the split GEOSYSTEMS batch files from the `\\middlman\sdrive\bats\GEO` directory to the newly created `D:\GEOgsd` directory.

   There will be 62 total GEOSYSTEMS batch files in the `\\middlman\sdrive\bats\GEO` directory. You will only want to use the 52 state batch files which there is GEOSYSTEMS street segment data for. Do not use the batch files for the 10 territories and military bases:

   AA, AE, AP, AS, FM, GU, MH, MP, PW, VI.
10. Create and edit a batch file that will call each of the individual GEOSYSTEMS state batch files and pass those batch files the required command line arguments. An example of this batch file is below:

    Example of a GEOSYSTEMS state GSD MKEM.BAT file

    ```
    call GEO_CO x:\georaw d: d:
    call GEO_CT x:\georaw d: d:
    call GEO_DC x:\georaw d: d:
    call GEO_DE x:\georaw d: d:
    call GEO_FL x:\georaw d: d:
    call GEO_GA x:\georaw d: d:
    call GEO_HI x:\georaw d: d:
    call GEO_IA x:\georaw d: d:
    call GEO_ID x:\georaw d: d:
    ```
11. The command line arguments for the GEOSYSTEMS state batch files are listed below:

    `X:\georaw`→location of the GEOSYSTEMS raw data files (%1 variable in the GEOSYSTEMS sample batch file).

    `D:`→local location of current `GSDLOAD\VERMMYY` directory (%2 variable in the GEOSYSTEMS sample batch file).

`D:`→local location of the USPS US.GSD file (%3 variable in the GEOSYSTEMS sample batch file).

12. Look through one of the GEOSYSTEMS state batch files to make certain that dates and directory names are correct and the directories referenced in the file actually exist and have the files needed contained in them. The important directories for this process and other information are given below:
    - Correct GSDRELEASEDATE.
    - Location of the GEOSYSTEMS raw data files.
    - Location of the USPS US.GSD file.
    - Location of the GSD log file.
    - Location of the current build utilities (verMMYY).
    - Location for the placement of the GEOSYSTEMS centroid files.
13. Make sure that the 5Zip.dbf file is contained in the GEOSYSTEMS centroid directory. This is normally in the `\\middleman\sdrive\GEOcent` directory.
14. After making sure that all of the needed utilities and data files are located where they should be and that all of the needed directories are made, run the MKEM.BAT batch file.

    NOTE: The first state you start will give you a message that the GEOSYSTEMS log file does not exist and should it create it. Say yes to this message and then restart the entire process, as the state that was being worked on when the message came up will have failed. Make sure the directory that was created for the state is deleted before restarting.
15. Perform the same operation on all of the other build computers so that all of the 52 GEOSYSTEMS state batch files get processed.
16. If you are running 6 or 7 files per build computer, the entire process should take a couple of days.
17. The output from this process is 52 primary state.gsd files, 52 supplemental state.gsd files and 52 state.zip back-up files. The state zip files contain the state primary and supplemental GSD files, logs and intermediate DBF files.

Sample batch file used to build a single state .gsd file

```
REM State Processing ALL PASSES
REM %1 = Geosystems RAW DATA DRIVE
REM D:    or    S:\GEORAW
REM %3 = DRIVE for US USPS file
REM %2 = LOCAL DRIVE
REM --------------------------------------
SET GSDRELEASEDATE=20000215
SET GSDLOGFILE=S:\GSDLOGS\GEO0200.DBF
cd \
cd \GEOSYS
MKDIR CA
cd CA
MKDIR TGR
CD TGR
copy %1\TGR06.ZIP
pkunzip TGR06
IF ERRORLEVEL 1 GOTO ERRTRAP
del TGR06.ZIP
```

```
IF ERRORLEVEL 1 GOTO ERRTRAP
CD ..
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0200\PASS1 TGR . %3\USPS0200\US;%2\GSDLOAD\VER0200 CA /GS
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0200\PASS2 . CA %2\GSDLOAD\VER0200
IF ERRORLEVEL 1 GOTO ERRTRAP
del 061.DBF
copy s:\usps0200\CA\CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
pkunzip CAZ4 06Z4.DBF
IF ERRORLEVEL 1 GOTO ERRTRAP
del CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0200\PASS3 . 06 . %2\GSDLOAD\VER0200 /N /C
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0200\PASS3 . 06 . %2\GSDLOAD\VER0200 /N /T
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0200\ZIPCENT . S:\GEOCENT CA
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER0200\PASS4 . CA S:\GEOCENT
IF ERRORLEVEL 1 GOTO ERRTRAP
REM START CLEAN UP
del *.CDX
del ??Z4.dbf
pkzip GEO_CA *.*
IF ERRORLEVEL 1 GOTO ERRTRAP
del ??P2.dbf
del ??_SHPE
del 06CENT.DBF
GOTO EXIT
:ERRTRAP
ECHO CA has had a failure > ERROR.TXT
:EXIT
cd \GEOSYS
REM ---------------------------------------------------
```

Merging state .gsd files into the us.gsd

NOTE: *MMYY stands for the current 2-digit month and 2-digit year.*

1. On the computers that ran the GEOSYSTEMS state batch files create a batch file called MVEM.BAT in the `D:\geogsd` directory.

   Example of mvem.bat batch file

   ```
   for /r %%f in (??.gsd) do move %%f s:\geo0400\state
   for /r %%f in (??t.gsd) do move %%f s:\geo0400\sup
   for /r %%f in (*.zip) do move %%f z:\geo04bck
   ```

   This batch file will recursively search through all of the sub-directories of the `D:\geogsd` directory for state primary or supplemental state GSD files as well as any state zip files. It will then move the primary GSD files to a single location, the supplemental GSD files to a separate location, and the State zip files to a back-up directory. The previous example is for the April 2000 build. We normally move all of the state GSD files to `\\middleman\sdrive` so that they are accessible from any of the other build computers. The zip back-up files are normally stored on `\\lizard\zdrive`.

2. Run MVEM.BAT on one build computer at a time
3. Count the number of primary state GSD files in the `\\middleman\sdrive\geoMMYY\state` directory; there should be 52. If there are less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case go to the computer that ran the specific state and in that state's build directory, under `D:\geogsd`, look at the logs to determine how it failed.
4. Count the number of supplemental state GSD files in the `\\middleman\sdrive\geoMMYY\sup` directory; there should be 52. If there are less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case, go to the computer that ran the specific state and in that state's build directory, under `D:\geogsd`, look at the logs to determine how it failed.
5. Copy all the GSD files from the `\\middleman\sdrive\uspsMMYY\other` directory (where MMYY is the same as `\\middleman\sdrive\geoMMYY`) into the `\\middleman\sdrive\geoMMYY\state` directory. Now the total primary state GSD files in the state directory should be the needed 62. If there are less check everything as in step 4 above and check to see that the `\\middleman\sdrive\uspsMMYY\other` directory has the required 10 USPS data only GSD files:

   AA.GSD, AE.GSD, AP.GSD, AS.GSD, FM.GSD, GU.GSD, MH.GSD, MP.GSD, PW.GSD, VI.GSD
6. In the root of the local D: drive, create a new directory called `geoMMYY.`
7. Also in the root of the local D: drive, create 2 other directories called `geoext` and `geoalts.`
8. Create 3 sub-directories under the `geoMMYY` directory called `state`, `sup` and `build.`
9. Copy ctystate.txt from the `\\middleman\sdrive\gsdload\verMMYY\ctyst` directory into the `D:\geoMMYY` directory.
10. Copy the current CTYST.DIR, MSAC.DIR and PARSE.DIR files from the `\\middleman\sdrive\gsdload\verMMYY` directory into the `D:\geoMMYY` directory.
11. In the `D:\geoMMYY\state` directory, copy the 62 state GSD files from the `\\middleman\sdrive\geoMMYY\state` directory.
12. In the `D:\geoMMYY\build` directory, copy the 62 building DBF files from `\\middleman\sdrive\buildMM` directory. These files are named in the following format: ??bldng.dbf, where ?? stands for the corresponding state, territory or military base's FIPS code. There should never be less or more than 62 of these building DBF files.
13. In the `D:\geoMMYY\sup` directory, copy the 52 supplemental GSD files from the `\\middleman\sdrive\geoMMYY\sup` directory.
14. In the `D:\geoext` directory, copy the contents of the `\\middleman\sdrive\gsdload\verMMYY` directory.
15. In the `D:\geoMMYY` directory, edit the MKEM.BAT batch file.
    - Uncomment all lines.
    - Change GSDRELEASEDATE to the correct date.
    - Make sure directories and files exist.

- Make sure the data type specified for the GSDMERGE.EXE utility is correct.
- Make sure that the locations of the state primary and supplemental GSD and building DBF file locations are correct for the GSDMERGE.EXE and BUILDING.EXE utilities.

16. Also in the `D:\geoMMYY` directory, edit the UNIQ.BAT batch file.
    - Make sure that the location of the state Z4 zip files is correct for the current build.
    - You will have to change the directory location for all 62 state, territory, and military base locations.
17. In the `D:\geoext` directory, edit the RUNALIAS.BAT batch file.
    - Make sure all directories and files exist.
    - Make sure the location of the GEOSYSTEMS state zip backup files is correct for the current build.
    - Make sure that the locations of US.GSD and UST.GSD are correct.

    NOTE: US.GSD and UST.GSD will be created prior to the call to the runalias.bat file.
18. Recheck the existence and locations of all directories and files, then run MKEM.BAT.

Example of mkem.bat

```
SET GSDRELEASEDATE=20000415
call uniq .\state\
gsdmerge .\state US.GSD GEOSYS /nad83
building .\build US.GSD US.GSD .
alias ctystate.txt US.GSD US.GSD .
gsdmerge .\sup UST.GSD GEOSYS /s /nad83
cd \geoext
call runalias
```

Example of uniq.bat

This is an example of only one state out of the 62 state, territory and military bases. This same example is repeated, in the UNIQ.BAT file, for each of the 62 locations.

```
pkunzip S:\USPS0400\AZ\AZZ4
uniquez4 04Z4.DBF CTYST.DIR PARSE.DIR %1AZ.GSD
del 04Z4.DBF
del 04ALT.DBF
```

Example of runalias.bat

```
cd \geoalts
for %%f in (z:\geo04bck\*.zip) do pkunzip %%f ??A.DBF
cd \geoext
spalias d:\geoalts .;d:\geo0400 /fix
spalias d:\geoalts d:\geo0400\us.gsd us.gsi .;d:\geo0400
copy us.gsi us_P.gsi
spalias d:\geoalts d:\geo0400\ust.gsd us.gsi .;d:\geo0400
copy us.gsi us_PS.gsi
stintbld d:\geo0400\us.gsd d:\geo0400\ust.gsd d:\geo0400\;.
us.gsi
```

```
for %%f in (z:\gdt
```

us_p.gsi

Relationships/dependencies upon other files

The us.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd as an input file, and the spalias utility.

us_ps.gsi

Relationships/dependencies upon other files

The ust.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using ust.gsd as an input file, and the spalias utility.

us.gsi

Relationships/dependencies upon other files

The us.gsd and ust.gsd files for this specific data type need to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd and ust.gsd as input files, and the stintbld utility.

ust.gsd

Relationships/dependencies upon other files

This file is composed of the 52 supplemental state .gsd files, which are then merged into ust.gsd by the mkem.bat file.

Build procedure

This file is built by merging the 52 state supplemental .gsd files using the gsdmerge utility with the \s flag. This file is built as part of the merge process for us.gsd; the file does not need to be built separately.

us.z9

Relationships/dependencies upon other files

The only difference between a US.Z9 for the different data types is the centroid files used as input to this process. For example, if building the GEOSYSTEMS US.Z9 then the input for this process would be the GEOSYSTEMS centroid files produced during the GEOSYSTEMS state GSD file build process.

Build procedure

??? stands for one of the street segment data types. GDT→GDT, TGR→ TIGER, NAV→Navigation Technologies, GEO→Geosystems, etc.

Steps 6 and 7 are performed because these are known data problems. More data issues could come up in the future but right now this is all we know about with the Centroid files.

1. On the local build computer’s D: drive, create a directory called `???cent` for the type of US.Z9 that is being built.
2. Create a batch file called MKCENTS.BAT (an example of this batch file follows this section). This batch file sets the GSDRELEASEDATE environment variable, then runs the ZIP4LOAD.EXE utility from the current `GSDLOAD` directory (make sure the directory name is correct for the current build) to build the US.Z9 file from the current build’s centroid files (make sure that the centroid file location is correct for the current build). The ZIPINDEX.EXE utility file is then used to index the newly created US.Z9.
3. In the data types centroid directory, normally `\\middlman\sdrive\???cent`, create a sub-directory called `7zip.`
4. In the `\\middlman\sdrive\???cent` directory, move all of the 7zip??.dbf files into the `7zip directory` (?? stands for the corresponding state’s FIPS code).

   NOTE: We do this because we do not want to build the US.Z9 with the 7-digit centroid files, only the 5- and 9-digit centroid files. We output the 7-digit centroid files so that we have the option.
5. Delete *.idx (index file) from the `\\middlman\sdrive\???cent` directory.
6. Open the 9zip26.dbf file (9-digit centroid file for MI).
   - Index the file based on the ZIP column.
   - Search for ZIP code 49950.
   - If the BlockGroup column contains a census tract of all 0’s change it to 26083960300 for all census tracts that need it.
   - The format of the Census tract ID is:

     SSCCCTTTT.TT

     Where SS=State Code

     CCC=County Code

     TTTT.TT=Census Tract
7. Open the 9zip02.dbf file (9=digit centroid file for AK)
   - Index the file based on the ZIP column
   - Search for ZIP code 99761

  - — Check to make sure that the CCC code is 185 and not 050
- Search for ZIP code 99636
  - — Check to make sure that the CCC code is 070 and not 050
- Search for ZIP code 99780
  - — Check to make sure that the CCC code is 261 and not 240

8. In the `D:\???cent` directory, run the MKCENTS.BAT file.

Sample batch file used for build

Example of mkcents.bat file

```
SET GSDRELEASEDATE=20000415
d:\gsdload\ver0400\zip4load US.Z9 s:\tgr cent
d:\gsdload\ver0400\zipindex US.Z9
```

Data files built from NAVTECH data

us.gsd

Relationships/dependencies upon other files

The us.gsd file is created by first building the individual state .gsd files, and then merging all the state .gsd files into the us.gsd file. The build procedure is almost exactly the same as for TIGER. The main thing is dir names are different.

Before the state .gsd build procedure is done, raw NAVTECH data needs to be converted to a TIGER file format, which is done using the nav2tgr.exe utility. Bill and George are the only ones who have ever used this utility.

Build procedure for state .gsd files

NAVTECH data only covers the 50 US states as well as D.C. and Puerto Rico for a total of 52. The remaining 10 territories and military bases are comprised of only USPS data but are still merged into the final US.GSD since they contain needed USPS information.

1. The first step to perform is to copy the USPS data only US.GSD from the `\\middleman\sdrive\uspsMMYY\us` directory to D:\ on the local computer. Normally we copy the USPS US.GSD into the following directory:

   `D:\uspsMMYY\us`

   NOTE: You can place this where you want it, but make sure you modify the batch files to reference this location. Refer to the example batch file after this section for more information.

2. Load RR (DOS Report Writer) from the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Retrieve the NAV Process Pass1 – ALL report. Edit this report for the current GSDRELEASEDATE, current `USPSMMYY` directories, current GSDLOAD version, etc. An example of this batch file follows this section.

3. In RR, output changed batch file to an ASCII file in the `\\middleman\sdrive\gsdload\verMMYY\bats` directory. Name it NAVALL.BAT. When RR outputs this file it will contain a copy of the exam-

ple batch file for each state, territory and military base. There are 52 total when building a NAVTECH street segment data US.GSD. The extra 2 states are for Puerto Rico and the District of Colombia.

NOTE: When merging any us.gsd file remember there has to be 62 state .gsd files, not 52. If you have any less than this then you are missing some.

4. Go to the `\\middlman\sdrive\bats\NAV` directory and del *.* from the directory.
5. Copy the NAVALL.BAT file to the `\\middleman\sdrive\bats\NAV` directory.
6. Use the FILSPLIT.EXE utility to split the NAVALL.BAT file into 62 state batch files (this utility is stored in the `C:\utils\system` directory of each build computer and is set in the environment PATH variable).

   Command line: `localdrive:\>filsplit navall.bat`

   FILSPLIT.EXE reads the file in and splits the file based on the individual states it encounters in the file. The output will be 62 batch files named as follows:

   NAV_AZ.BAT (for the state of Arizona)
7. On the local build computer, create a directory called `navgsd` in the root of D:\.
8. Determine how many build computers are available for processing. You should always try to have 8 available. This way you can split up the processing load evenly across multiple build computers
9. Copy the pre-determined number of the split NAVTECH batch files from the `\\middlman\sdrive\bats\NAV` directory to the newly created `D:\navgsd` directory.

   There will be 62 total NAVTECH batch files in the `\\middlman\sdrive\bats\nav` directory. You will only want to use the 52 state batch files which there is NAVTECH street segment data for. Do not use the batch files for the 10 territories and military bases:

   AA, AE, AP, AS, FM, GU, MH, MP, PW, VI.
10. Create and edit a batch file that will call each of the individual NAVTECH state batch files and pass those batch files the required command line arguments. An example of this batch file is below:

    Example of a NAVTECH state GSD MKEM.BAT file

    ```
    call NAV_CO x:\navraw d: d:
    call NAV_CT x:\navraw d: d:
    call NAV_DC x:\navraw d: d:
    call NAV_DE x:\navraw d: d:
    call NAV_FL x:\navraw d: d:
    call NAV_GA x:\navraw d: d:
    call NAV_HI x:\navraw d: d:
    call NAV_IA x:\navraw d: d:
    call NAV_ID x:\navraw d: d:
    ```
11. The command line arguments for the NAVTECH state batch files are listed below:

    X:\navraw →location of the NAVTECH raw data files (%1 variable in the NAVTECH sample batch file).

    `D:`→local location of current `GSDLOAD\VERMMYY` directory (%2 variable in the NAVTECH sample batch file).

`D:`→local location of the USPS US.GSD file (%3 variable in the NAVTECH sample batch file).

12. Look through one of the NAVTECH state batch files to make certain that dates and directory names are correct and the directories referenced in the file actually exist and have the files needed contained in them. The important directories for this process and other information are given below:
    - Correct GSDRELEASEDATE.
    - Location of the NAVTECH raw data files.
    - Location of the USPS US.GSD file.
    - Location of the GSD log file.
    - Location of the current build utilities (verMMYY).
    - Location for the placement of the NAVTECH centroid files.
13. Make sure that the 5Zip.dbf file is contained in the NAVTECH centroid directory. This is normally in the `\\middleman\sdrive\navcent` directory.
14. After making sure that all of the needed utilities and data files are located where they should be and that all of the needed directories are made, run the MKEM.BAT batch file.

    NOTE: The first state you start will give you a message that the NAVTECH log file does not exist and should it create it. Say yes to this message and then restart the entire process, as the state that was being worked on when the message came up will have failed. Make sure the directory that was created for the state is deleted before restarting.
15. Perform the same operation on all of the other build computers so that all of the 52 NAVTECH state batch files get processed.
16. If you are running 6 or 7 files per build computer, the entire process should take a couple of days.
17. The output from this process is 52 primary state.gsd files, 52 supplemental state.gsd files and 52 state.zip back-up files. The state zip files contain the state primary and supplemental GSD files, logs and intermediate DBF files.

Sample batch file used to build a single state .gsd file

```
REM State Processing ALL PASSES
REM %1 = Geosystems RAW DATAT DRIVE
REM D:    or    S:\GEORAW
REM %3 = DRIVE for US USPS file
REM %2 = LOCAL DRIVE
REM ---------------------------------------
SET GSDRELEASEDATE=19991215
SET GSDLOGFILE=S:\GSDLOGS\NAV1299.DBF
cd \
cd \NAVGSD
MKDIR CA
cd CA
MKDIR TGR
CD TGR
IF ERRORLEVEL 1 GOTO ERRTRAP
copy %1\NAV_CA.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
pkunzip NAV_CA
IF ERRORLEVEL 1 GOTO ERRTRAP
del NAV_CA.ZIP
```

```
IF ERRORLEVEL 1 GOTO ERRTRAP
CD ..
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER1299\PASS1 TGR . %3\USPS1299\US;%2\GSDLOAD\VER1299 CA /4
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER1299\PASS2 . CA %2\GSDLOAD\VER1299
IF ERRORLEVEL 1 GOTO ERRTRAP
del 061.DBF
copy s:\usps1299\CA\CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
pkunzip CAZ4 06Z4.DBF
IF ERRORLEVEL 1 GOTO ERRTRAP
del CAZ4.ZIP
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER1299\PASS3 . 06 . %2\GSDLOAD\VER1299 /N /C
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER1299\PASS3 . 06 . %2\GSDLOAD\VER1299 /N /T
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER1299\ZIPCENT . S:\NAVCENT CA
IF ERRORLEVEL 1 GOTO ERRTRAP
%2\GSDLOAD\VER1299\PASS4 . CA S:\NAVCENT
IF ERRORLEVEL 1 GOTO ERRTRAP
REM START CLEAN UP
del *.CDX
del ??Z4.dbf
pkzip NAV_CA *.*
IF ERRORLEVEL 1 GOTO ERRTRAP
del ??P2.dbf
del ??_SHPE
del 06CENT.DBF
GOTO EXIT
:ERRTRAP
ECHO CA has had a falure > ERROR.TXT
:EXIT
cd \NAVGSD
REM ---------------------------------------------------
```

Merging state .gsd files into the us.gsd

NOTE: *MMYY stands for the current 2-digit month and 2-digit year.*

1. On the computers that ran the NAVTECH state batch files create a batch file called MVEM.BAT in the `D:\navgsd` directory.

   The following is an example of the mvem.bat batch file:

   ```
   for /r %%f in (??.gsd) do move %%f s:\nav0400\state
   for /r %%f in (??t.gsd) do move %%f s:\nav0400\sup
   for /r %%f in (*.zip) do move %%f z:\nav04bck
   ```

   This batch file will recursively search through all of the subdirectories of the `D:\navgsd` directory for state primary or supplemental state GSD files as well as any state zip files. It will then move the primary GSD files to a single location, the supplemental GSD files to a separate location, and the state zip files to a back-up directory. The previous example is for the April 2000 build. We normally move all of the state GSD files to `\\middleman\sdrive` so that they are accessible from any of the other build computers. The zip back-up files are normally stored on `\\lizard\zdrive.`

2. Run MVEM.BAT on one build computer at a time
3. Count the number of primary state GSD files in the `\\middleman\sdrive\navMMYY\state` directory; there should be 52. If there are less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case go to the computer that ran the specific state and in that state's build directory, under `D:\navgsd`, look at the logs to determine how it failed.
4. Count the number of supplemental state GSD files in the `\\middleman\sdrive\navMMYY\sup` directory; there should be 52. If there are less then you missed one either by not running it or moving it with the MVEM.BAT file or the missing state's batch file failed. If the latter is the case, go to the computer that ran the specific state and in that state's build directory, under `D:\navgsd`, look at the logs to determine how it failed.
5. Copy all the GSD files from the `\\middleman\sdrive\uspsMMYY\other` directory (where MMYY is the same as `\\middleman\sdrive\navMMYY`) into the `\\middleman\sdrive\navMMYY\state` directory. Now the total primary state GSD files in the state directory should be the needed 62. If there are less check everything as in step 4 above and check to see that the `\\middleman\sdrive\uspsMMYY\other` directory has the required 10 USPS data only GSD files:

   AA.GSD, AE.GSD, AP.GSD, AS.GSD, FM.GSD, GU.GSD, MH.GSD, MP.GSD, PW.GSD, VI.GSD
6. In the root of the local D: drive, create a new directory called `navMMYY.`
7. Also in the root of the local D: drive, create 2 other directories called `navext` and `navalts.`
8. Create 3 sub-directories under the `navMMYY` directory called `state`, `sup,` and `build.`
9. Copy ctystate.txt from the `\\middleman\sdrive\gsdload\verMMYY\ctyst` directory into the `D:\navMMYY` directory.
10. Copy the current CTYST.DIR, MSAC.DIR and PARSE.DIR files from the `\\middleman\sdrive\gsdload\verMMYY` directory into the `D:\navMMYY` directory.
11. In the `D:\navMMYY\state` directory, copy the 62 state GSD files from the `\\middleman\sdrive\navMMYY\state` directory.
12. In the `D:\navMMYY\build` directory, copy the 62 building DBF files from `\\middleman\sdrive\buildMM` directory. These files are named in the following format: ??bldng.dbf, where ?? stands for the corresponding state, territory or military base's FIPS code. There should never be less or more than 62 of these building DBF files.
13. In the `D:\navMMYY\sup` directory, copy the 52 supplemental GSD files from the `\\middleman\sdrive\navMMYY\sup` directory.
14. In the `D:\navext` directory, copy the contents of the `\\middleman\sdrive\gsdload\verMMYY` directory.
15. In the `D:\navMMYY` directory, edit the MKEM.BAT batch file.
    - Uncomment all lines.
    - Change GSDRELEASEDATE to the correct date.

- Make sure directories and files exist.
- Make sure the data type specified for the GSDMERGE.EXE utility is correct.
- Make sure that the locations of the state primary and supplemental GSD and building DBF file locations are correct for the GSDMERGE.EXE and BUILDING.EXE utilities.

16. Also in the `D:\navMMYY` directory, edit the UNIQ.BAT batch file.
    - Make sure that the location of the state Z4 zip files is correct for the current build.
    - You will have to change the directory location for all 62 state, territory, and military base locations.
17. In the `D:\navext` directory, edit the RUNALIAS.BAT batch file.
    - Make sure all directories and files exist.
    - Make sure the location of the NAVTECH state zip backup files is correct for the current build.
    - Make sure that the locations of US.GSD and UST.GSD are correct.

    NOTE: US.GSD and UST.GSD will be created prior to the call to the runalias.bat file.
18. Recheck the existence and locations of all directories and files, then run MKEM.BAT.

Example of mkem.bat

```
SET GSDRELEASEDATE=20000415
call uniq .\state\
gsdmerge .\state US.GSD NAV /nad83
building .\build US.GSD US.GSD .
alias ctystate.txt US.GSD US.GSD .
gsdmerge .\sup UST.GSD NAV /s /nad83
cd \navext
call runalias
```

Example of uniq.bat

This is an example of only one state out of the 62 state, territory and military bases. This same example is repeated, in the UNIQ.BAT file, for each of the 62 locations.

```
pkunzip S:\USPS0400\AZ\AZZ4
uniquez4 04Z4.DBF CTYST.DIR PARSE.DIR %1AZ.GSD
del 04Z4.DBF
del 04ALT.DBF
```

Example of runalias.bat

```
cd \navalts
for %%f in (z:\nav04bck\*.zip) do pkunzip %%f ??A.DBF
cd \navext
spalias d:\navalts .;d:\nav0400 /fix
spalias d:\navalts d:\nav0400\us.gsd us.gsi .;d:\nav0400
copy us.gsi us_P.gsi
spalias d:\navalts d:\nav0400\ust.gsd us.gsi .;d:\nav0400
copy us.gsi us_PS.gsi
```

```
stintbld d:\nav0400\us.gsd d:\nav0400\ust.gsd d:\nav0400\;.
us.gsi
for %%f in (z:\gdt
```

us_p.gsi

Relationships/dependencies upon other files

The us.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd as an input file, and the spalias utility.

us_ps.gsi

Relationships/dependencies upon other files

The ust.gsd for this specific data type needs to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using ust.gsd as an input file, and the spalias utility.

us.gsi

Relationships/dependencies upon other files

The us.gsd and ust.gsd files for this specific data type need to be created first.

Build procedure

This file is built as part of the merge process for us.gsd; the file does not need to be built separately. It is actually built in runalias.bat, using us.gsd and ust.gsd as input files, and the stintbld utility.

ust.gsd

Relationships/dependencies upon other files

This file is composed of the 52 supplemental state .gsd files, which are then merged into ust.gsd by the mkem.bat file.

Build procedure

This file is built by merging the 52 state supplemental .gsd files using the gsdmerge utility with the \s flag. This file is built as part of the merge process for us.gsd; the file does not need to be built separately.

us.z9

Relationships/dependencies upon other files

The only difference between a US.Z9 for the different data types is the centroid files used as input to this process. For example, if building the NAVTECH US.Z9 then the input for this process would be the NAVTECH centroid files produced during the NAVTECH state GSD file build process.

Build procedure

??? stands for one of the street segment data types. GDT→GDT, TGR→ TIGER, NAV→Navigation Technologies, GEO→Geosystems, etc.

Steps 6 and 7 are performed because these are known data problems. More data issues could come up in the future but right now this is all we know about with the Centroid files.

1. On the local build computer's D: drive, create a directory called `???cent` for the type of US.Z9 that is being built.
2. Create a batch file called MKCENTS.BAT (an example of this file is given after this section).This batch file sets the GSDRELEASEDATE environment variable, then runs the ZIP4LOAD.EXE utility from the current `GSDLOAD` directory (make sure the directory name is correct for the current build) to build the US.Z9 file from the current build's centroid files (make sure that the centroid file location is correct for the current build). The ZIPINDEX.EXE utility file is then used to index the newly created US.Z9.
3. In the data types centroid directory, normally `\\middlman\sdrive\???cent`, create a sub-directory called `7zip.`
4. In the `\\middlman\sdrive\???cent` directory, move all of the 7zip??.dbf files into the `7zip` directory (?? stands for the corresponding state's FIPS code).

   NOTE: We do this because we do not want to build the US.Z9 with the 7-digit centroid files, only the 5- and 9-digit centroid files. We output the 7-digit centroid files so that we have the option.
5. Delete *.idx (index file) from the `\\middlman\sdrive\???cent` directory.
6. Open the 9zip26.dbf file (9-digit centroid file for MI).
   - Index the file based on the ZIP column.
   - Search for ZIP code 49950.
   - If the BlockGroup column contains a census tract of all 0s change it to 26083960300 for all census tracts that need it.
   - The format of the Census tract ID is:

     SSCCCTTTT.TT

     Where SS=State Code

     CCC=County Code

     TTTT.TT=Census Tract
7. Open the 9zip02.dbf file (9=digit centroid file for AK)
   - Index the file based on the ZIP column
   - Search for ZIP code 99761

   - — Check to make sure that the CCC code is 185 and not 050
- Search for ZIP code 99636
   - — Check to make sure that the CCC code is 070 and not 050
- Search for ZIP code 99780
   - — Check to make sure that the CCC code is 261 and not 240

8. In the `D:\???cent` directory, run the MKCENTS.BAT file.

Sample batch file used for build

Example of mkcents.bat file

```
SET GSDRELEASEDATE=20000415
d:\gsdload\ver0400\zip4load US.Z9 s:\nav cent
d:\gsdload\ver0400\zipindex US.Z9
```

Data files built from Etak data

The file build procedures are very similar to those for TIGER, GDT, and NAVTECH. The only difference is that different street segment input files from different vendors are used.

Custom file builds

Ameritech files

1. Copy Amtec directory on s:\ to the Amtec directory on the local drive.
2. Change the path on s:\ to be the path to the `gdt(mim?)\7zip` directory.
   - In `7zip`, ’7’ is under the 9zip directory.
   - In `7zip`, "zip" contains 7zip??.dbf (?? is the state abbreviation) output from the us.z9 build.
3. Run mkst.prg in FoxPro.

   Mkst.prg makes zip7??.dbf (?? = st_anon?) files from 7zip??.dbf (?? = state number) files and grabs only zip7, lon, and lat columns.
4. Run mkem.prg in FoxPro.

   Mkem.prg dumps all zip7??.dbf files into uszip7.dbf, based on zip7str.dbf.
5. Move uszip7.dbf to a place where it will be used.

Claratlas files by BG, ZIP+4, and Demographics

BG + Prizm codes—indexed by BG

1. Copy bgprizm.dbf from the Claratlas CD.
2. Edit bgprizm.cfg (same as NHDB.cfg):
   - For record length + number of records + offset
   - For fields (name, description, offset)
   - For record offset = 0

3. Edit bgprizm.spc (same as NHDB.spc)
   - For data file name
   - CFG file location
   - Index directory

Z4 + Prizm codes—indexed by Z4

1. Open 1 zip4_??.dbf in FoxPro

   ?? is the state
2. Copy the structure to temp.dbf (zip4, Prizm codes)
3. Use temp.dbf and append (place all above into a .prg file to run in FoxPro) from all zip4_??.dbf Claratas files.
4. Run .prg
5. Edit z4prizm.cfg, z4prizm.spc (changes field names, record length, etc.).

   File names = z4prizm

BG + Demographics—indexed by BG

1. Open fields.dbf in FoxPro
2. Edit Claratlas file in FoxPro so that the extended structure looks like fields.dbf (copy the extended structure to file).
3. Edit bgdemos.cfg (same as NHDB.cfg) to match fields in Claratlas file.
4. Edit hgdemos.spc for the file name and .cfg path.

County and state GSBs

1. Install Atlas GIS
2. In the Atlas directory `\Data` , there are ak.agf, hi.agf, and counties.agf files.
3. Load Atlas.
4. Load two .agf files.
5. Delete Alaska and Hawaii from counties.agf.
6. Turn of all layers but counties. In the next round, turn of all layers but states.
7. Merge into one layer (counties.agf + ak.agf/hi.agf).
8. Export as a MID/MIF file.
9. Import a MID/MIF file into Centrus Desktop as a .GSB file.

Dontec files

1. Change directory to `s:\gdtcent`. This is where the current 9zip??.dbf files are located.
2. Copy the .dbf files for Iowa, Indiana, and Illinois into the Dontec directory where they are accessable.

   Files are named as follows:
   - 9zip??.dbf (?? = state FIPS code)

3. Rename the proper file from 9zip??.dbf (?? = FIPS code) to 9zip??.dbf (?? = state abbreviation).

   IL = 17

   IN = 18

   IA = 19

Experian, Progressive, and Allstate files

Build procedure

The Progressive data build is predicated upon the Allstate and Experian builds (that is, you must build the Allstate and Experian files before building the Progressive files).

NOTE: *All of the needed FoxPro programs and files are in the \\thistle\D\dat directory.*

1. Copy the current month's GDT US.Z9 into the `\\thistle\D\dat\progess` directory.
2. Run the mkem.bat file, located in the `\\thistle\D\dat\progess` directory.

   The following is an example of a mkem.bat file:

   ```
   Z9dump US.Z9 000010001 099999999 z0.DBF
   Z9dump US.Z9 100010001 199999999 z1.DBF
   Z9dump US.Z9 200010001 299999999 z2.DBF
   Z9dump US.Z9 300010001 399999999 z3.DBF
   Z9dump US.Z9 400010001 499999999 z4.DBF
   Z9dump US.Z9 500010001 599999999 z5.DBF
   Z9dump US.Z9 600010001 699999999 z6.DBF
   Z9dump US.Z9 700010001 799999999 z7.DBF
   Z9dump US.Z9 800010001 899999999 z8.DBF
   Z9dump US.Z9 900010001 999999999 z9.DBF
   ```

   The following is an example of a Z9dump.exe command line:

   ```
   Z9dump US.Z9 StartPoint EndPoint OutputFile
   ```

   This batch file uses the Z9dump.exe utility, located in the `\\thistle\D\dat\progess` directory, to create 10 .dbf files based on the first digit of all the possible zip codes (0-9).
3. In FoxPro, run the Zip4cnt.prg program. This program reads all of the z0-z9.dbf files, produced in step 2, and outputs Zip4cnt.dbf and Zipcnty.dbf. The Zip4cnt.dbf is a table that shows the number of ZIP+4s in each 5-digit zip code. The zipcnty.dbf is a table that shows the number of counties in each 5-digit zip code.
4. Experian gets the Zip4cnt.dbf file as well as a text version of this file. To make the text version, open the Zip4cnt.dbf file in FoxPro. In the command box type:

   ```
   Copy Zip4cnt to Zip4cnt.txt SDF
   ```

5. The Zipcnty.dbf file is used as input for the Allstate files.

6. Copy the Zipcnty.dbf file into the `\\thistle\D\dat\allstate` directory. Copy the current CTYST.DBF file, located in the `\\middleman\Sdrive\gsdload\verMMYY\ctyst` directory (where MMYY is the current build month and year), into the `\\thistle\D\dat\allstate` directory.
7. In FoxPro, run the Addpo.prg program. This program adds the missing ZIP codes and FIPS county codes to the Zipcnty.dbf file and appends all the corresponding post office names to the file. The missing postal information comes from the CTYST.DBF file. The output from this program is the fincnty.dbf file.
8. Copy the previous month's fincnty.dbf file into the `\\thistle\D\dat\allstate\mrg` directory. This file may already exist in this directory if you built the Allstate files in this directory during the previous month (it will be named in the following format: fin??.dbf where ?? stands for the 2-digit month. E.g., 03 for March). If you have to copy the previous month's fincnty.dbf into this directory, rename it to fin??.dbf.
9. Copy the current month's fincnty.dbf into the `\\thistle\D\dat\allstate\mrg` directory. Rename it to fin??.dbf.
10. Delete any .idx and .txt files from the `\\thistle\D\dat\allstate\mrg` directory. If the 2-month-old fin??.dbf file exists in the `\\thistle\D\dat\allstate\mrg` directory, delete it as well.
11. Edit the finmrg.prg program in a text editor. Ths file is located in the `\\thistle\D\dat\allstate\mrg` directory. Change all of the fin?? file names to the correct ones that are going to be used. For example, if the finmrg.prg file was used in the month of March, the files being used by it would be fin02 and fin03 (02 for February and 03 for March). If you were now building the Allstate files for April you would change all of the references of fin02 to fin03 and all references of fin03 to fin04‡fin04 would be to current fincnty.dbf which you already renamed.
12. Run FoxPro and execute finmgr.prg.
    - Finmgr.prg checks to see what ZIP codes have changed, been added, or been removed.
    - The output is zip??.txt (previous month), zip??.txt (current month), except.txt.
13. Delete except.dbf, zipadd.dbf, zipchange.dbf, and zipmove.dbf. Overwrite temp.dbf.

    Note: *This is the end of the Allstate and Experian builds. The rest of this procedure finishes the Progressive data build.*

14. In `G:\Allstate` on Bill Flanagan's machine, execute the following command line:

    ```
    mkdir mmMM
    ```

    mm = old month

    MM = new month
15. Copy three text files from `S:\DAO\MGR` to the new directory.
16. Run FoxPro and execute mktxt.prg.

17. Mktxt.prg takes ten .dbf files and converts them to text files (e.g. z0.dbf is converted to z0.txt). The .dbf files are deleted upon completion.
18. Zip the ten .txt files into a file called zip9.zip.
19. Copy zip9.zip to a directory where it can be easily accessed by the build master.

Note: *This completes the Progressive data build.*

Finmbr.dat

1. Copy (or move) x.dir to the destination directory.

   Destination directory = local drive (where .gsd was built) or `s:\gdt????`.
2. Copy (or move) x.gsd to the GDT destination directory.
3. Copy the `finmbr` directory from the current verison of GSDLOAD on s:\.
4. Run finbuild using the following command-line arguments:

   `finbuild` *`current_dir output_filename`*

Prizmz4.dbf

Note: *This procedure is current as of 05/08/00.*

Build procedure

To create the Claritas Prizmz4.dbf file using Claritas Prizm data, perform the following procedure.

1. Insert the latest version of the Claritas Prizm .dbf file CD labeled “1999 Prizm License Files for all 9 digit ZIP Codes” into the local CD-ROM drive.
2. Combine all (51) Claritas .dbf files into one .dbf file named prizmz4.dbf by following this subprocedure:
   - Get all files from the `PVCS\DataBuilds\Claritas\Prizm` project.
   - Verify the d: drive has at least 750 MB of free space for the .dbf merged file.
   - Open Foxpro.
   - Run the `d:\databuilds\claritas\prizmraw\prizmmerge.prg` program. This program does the following tasks:
     - — Copies the `d:\databuilds\claritas\prizmraw\prizmz4blank.dbf` file to `d:\databuilds\claritas\prizmraw\prizmz4.dbf` with no records.
     - — Appends the data from all 51 individual state .dbf files to prizmz4.dbf
     - — Lets you browse the file upon completion.
3. Verify the data in prizmz4.dbf is complete with all 51 states.
4. Modify the table structure of prizmz4.dbf to rename the first two fields’ names as follows:
   - Change the first field’s name from cyzip_4 to zip4.
   - Change the second field’s name from Clus_code to Prizm.

   These field names must be changed because the application executed in the next step doesn't know what the Claritas default fields mean.

5. Copy the file `d:\databuilds\claritas\prizmraw\prizmz4.dbf` to `d:\databuilds\claritas\middleman_s\demoload\prizm\prizmz4.dbf.`
6. Open a DOS prompt and change to the `d:\databuilds\claritas\middleman_s\demoload\prizm` directory.
7. Prepare to build the prizmz4.dld file by doing the following subprocedure:
    - Get all files from the `PVCS\DataBuilds\Claritas\middleman_s\demoload\Prizm` project.
    - Get all files from the PVCS `DataBuilds\Claritas\middleman_s\demoload\clar` project.
    - Verify the C: drive has at least 1.5 GB of free space. The dlbuild.exe app will write a temporary index to the temp directory while it is executing.
8. Execute `d:\databuilds\claritas\middleman_s\demoload\clar\dlbuild prizmz4.ini`.

    Look at the .ini file and make sure that the events that should happen, will happen.

    This step will produce the prizmz4.dld file as its output.
9. Verify that the dlbuild process finished without errors and that prizmz4.dld was created.
10. Test the resulting prizmz4.dld file for functionality.

    Note: This test has not yet been developed.
11. Deliver the prizmz4.dld file to Steve Walden and Henry Bequet for internal use.