CITYWIDE TRANSIT TRAVEL DATABASE

FINAL REPORT

Prepared for the MTA / NYCT

January 18, 2008

Prepared by:

Caliper Corporation
1172 Beacon Street Suite 300
Newton, MA 02461-1146

(617) 527-4700
http://www.caliper.com

Table of Contents

- Introduction/Overview 4
- Input Datasets 6
  - EU65 MetroCard Transactions 7
  - Bus Trip Table 9
  - NYCT Bus Schedules 12
  - Long Island Bus Schedules 16
  - Metro-North Hudson Rail-Link Bus Schedules 18
  - DOT Bus Schedules 18
  - Atlantic Express Bus Schedules 19
  - Subway Schedules 20
  - Staten Island Railway Schedules 21
  - JFK AirTrain Schedules 21
  - PATH Schedules 21
  - Roosevelt Island Tramway Schedules 22
  - Staten Island Ferry Schedules 22
  - Transit GIS Route System 22
  - Station List 27
  - Booths 28
  - Bus Depot List 29
  - DOT Box List 30
  - Tatmaster 30
  - GIS Reference Layers 32
  - CTPP Part 3 Data 34
  - Senior MetroCards 34
- Data Processing 35
  - Lookup Table Creation 37
  - AFC Bus Trip Processing 41
  - Processing of EU65 Transactions 43
    - Locating Subway Transactions 43
    - Locating Bus Transactions 43
    - Detailed Description of MetroCard Transaction Processing 45
- Trip Matrices 53
  - AM Peak Trip Matrix 53
  - AM Peak Subway Matrix 53
- Administrative Toolbox 54
  - Installation 54
- Validation 56
- Query Software 61
- Maintenance 64
- Concluding Remarks 65
- References 66

Citywide Transit Travel Database Final Report

This document constitutes the final report for the work performed by Caliper Corporation for the Citywide Transit Travel Database project. The goal of this project was to create a functioning software system that extracts transit trip information from New York City Transit's Automatic Fare Collection (AFC) system and combines it with other NYCT datasets to create a wealth of information about transit utilization in New York City. The principal information to be obtained was daily information on the origin and destination of trips utilizing the buses and subways. Importantly, the system creates the first good estimates of linked trips that have been available in a long time. Origin-destination information is also provided from travel analysis zone to travel analysis zone in addition to from stop to stop facilitating forecasting of utilization of major new services that are being evaluated. A dynamic portrait of utilization is provided as transit boardings and alightings can be estimated by time period and by route location. Even though the estimation process is not without error, the data produced by the system should be invaluable for service planning.

While this project involved significant software development, the most challenging aspects of the work involved developing a workable process for creating geo-located trip origins and destinations both on the transit system and between transportation analysis zones. From the outset, neither NYCT nor Caliper was certain that the origins and destinations of bus trips could be properly located due to data limitations and methodological obstacles. In fact, our first several attempts to solve this problem were met with failure. Nevertheless, through the collaboration of NYCT and Caliper staff, a workable and computable solution was finally obtained.

User friendly query software, developed for this project, allows the trips to be queried, summarized, visualized and exported. The query software includes the ability to create inputs for NYCT's Operations Planning Department (OP) transit trip assignment model and to examine origin and destination (O-D) patterns and trip length distributions of riders by individual routes for NYCT's Office of Management and Budget (OMB). The processing system created, built on top of Caliper's TransCAD GIS software, works around the limitations of the available data and efficiently handles the seven million transactions logged each day on the New York City Transit system

This report documents the processing procedures and software created for the project. It provides technical details of the input and output datasets, and the algorithms used. A discussion of the problems encountered along the way and how they were overcome is also included. This report describes the query software developed, so that custom data extracts can easily be created by NYCT and MTA users. These extracts can exported to standard formats or summarized in reports, maps, origin-destination matrices, etc. The report concludes with a discussion of the project's limitations and possible future research.

An Atlas of Transit Ridership and a User's Guide for the query software were produced for this project and are available separately.

Introduction/Overview

NYC's MetroCard system is an entry-only AFC system on which a rider swipes a MetroCard to enter a subway station or dips a MetroCard to board a bus, generating a unique transaction that is logged to a mainframe computer database. No transaction occurs when a rider exits a station or a bus. The system was designed in the early 1990s to collect fares efficiently from millions of riders per day with a flat rate fare structure. It was not setup to capture the details of the trips made by each rider. Saved transaction times are truncated to tenths of an hour, due to memory limitations in the vintage hardware.

Once AFC systems came into use, transportation planners realized that they contained a wealth of information that could be useful as inputs to operations planning and demand forecasting models. Early work by Barry, Newhouser, Rahbee and Sayeda [BNRS02], showed that MetroCard transactions could be used to estimate subway O-D patterns. Rahbee then moved to Chicago, where he applied similar techniques to CTA's system [RC02].

The major goal of this project was to extend [BNRS02]'s work to include all transit modes: subway, local and express busses, ferry and tramway. Other goals included improving the estimation process for subway trips, generating origin-destination patterns by traffic analysis zone, and generally streamlining all of the data processing and analysis that users might want.

The core of the approach is that for each transaction, we attempt to identify the route and the specific boarding and alighting stops that define a trip leg. We combine multiple trip legs into a linked trip, when it is inferred that a rider uses his/her MetroCard two or more times to complete a single journey.

Subway and tramway boardings are located using the turnstile fare collection information. This provides an immediate identification of the station, but not the route(s) boarded when multiple lines serve the same station complex or the rider switches subway trains.

Bus boardings are located by estimating the location of the bus at the trip boarding transaction time. The AFC Bus Trip table is combined with the bus schedules and positional information derived from certain MetroCard transactions to obtain approximate bus locations for most trips, interpolating using the distance between stops. The boarding locations developed are geographically less accurate than for subways, due to the six minute truncation of recorded boarding times, the estimation process, and the cleanup required to use the AFC Bus Trip table. The route can usually be determined from the bus trip table, except when the data are missing such as when the sign code was not updated correctly on a bus.

We make two assumptions, shown by [BNRS02] to be reasonable for subway riders that allow us to determine alighting locations:

1. Most riders start their next trip at or near the destination of their previous trip
2. Most riders end their last trip of the day at or near the start of their first trip of the day

We make an additional assumption that the pattern of single-fare card users is similar to that of multiple-fare card users at a given boarding location.

We apply a chaining procedure to determine the likely alighting locations for riders with two or more MetroCard transactions on a particular day. This assumes that many riders start their next movement near the conclusion of their prior movement. Impossible destinations, those that are unreachable by the subway system or bus in question, are discarded. Destinations for single trips and other trips with no chained destination are assigned using random sampling of distributions derived from other riders who have the same trip origins.

Two or more movements for a rider are linked together into a single trip, when they occur within a short period of time. The alighting times for subway trips are determined using TransCAD's schedule-based shortest path (SSP) algorithm, which utilizes the complete subway schedule and geographic representations of all route patterns, to predict the route traveled through the subway system and the time of arrival. This method also handles the use of the Staten Island Ferry. Bus alighting times are determined using the estimated arrival time for the bus at the alighting stop.

The procedure generates a table of linked passenger trips for a given day, each trip with an origin and destination that is either a subway station or bus stop. Nearby origin and destination zones (2000 Census Block Groups) were assigned to each trip, using a logit allocation procedure that distributes the trips to nearby zones based on a weighting of walking distance, and population and/or employment depending on the time of day. A second table details the component legs for each linked trip. Each leg record contains its mode, route, and origin and destination locations and times. These two tables are the principal inputs used by the query software.

The powerful query software allows almost any conceivable query to be answered; it works in two steps: trip/leg selection and output creation. Queries can be made on either the linked trips table or the unlinked legs table. The *Query Builder* step defines a query, combining one or more selection criteria that conceptually select a set of trips/legs. The query can require that either any or all of the primitives be matched. For linked trips, the criteria include:

- Selecting by the mode, route, or pattern of a particular leg in the trip sequence
- Selecting by the origin and/or destination of the trip by specifying stops, zones, Census Tracts or Boroughs
- Selecting by the inclusion of a particular type of transfer between modes within the trip; or by a general SQL query

For unlinked legs, the selection criteria include:

- Selecting by the mode, route or pattern of the leg
- Selecting by the origin and/or destination of the leg by specifying stops, zones, Census Tracts or Boroughs
- Selecting by specifying the mode of either the proceeding or following leg used as part of a linked trip
- Selecting by a general SQL query
- Selecting by the position of the leg within its linked trip

Queries can be saved for later reuse.

The *Execute* step specifies the day to use for the query, selected from the list of available days. The trips/legs can be further restricted based on a time period and/or an existing TransCAD selection set of linked trips. The choices for the output are numerous and include:

- Reports, summarized by combinations of
  - Arrival time
  - Departure time
  - Mode or route of a particular leg
  - Origin
  - Destination
  - Origin-Destination pair
- A ridership report by either route pattern or route segment
- Maps that depict the origins and/or destinations of the trips/legs selected. The maps can also include a scaled theme depicting the ridership by street/track segment.
- O-D matrices summarizing the trips/legs by
  - Stop
  - Zone
  - Census Tract
  - Borough

  The rows and columns can be summarized by different levels of geography.
- A TransCAD selection set that can be used to create external tables/spreadsheets. Besides creating export tables, the selection set can be used for general analysis in TransCAD or for examining individual linked trips in a customized trip browser that depicts each leg of the linked trip on the map.

In summary, this project involved creating custom software that processes MetroCard data and creates geo-located linked passenger trips. These trips can then be queried using user-friendly software to produce reports, maps, extracts and matrices as needed to support various planning and operational needs. Numerous unexpected hurdles were overcome to create a functioning system that we believe is the first to actually handle buses and mixed-mode trips for a transit system with entry-only AFC data.

Input Datasets

This section documents the various input datasets used by the Transit Travel Database project. For each dataset, we describe its source, contents and how it was processed into a usable format. We also provide descriptive summary statistics for some of the datasets.

The principal datasets contain transactional data from a two week study period from April 19 to May 3, 2004 (3am–3am) that was selected as a representative period for data collection and was free from exceptional events, such as snow storms, disasters and school vacations. These datasets include the EU65 transactions and the bus trip logs collected from the MetroCard system. The data are augmented with a variety of reference datasets, including schedules, a GIS-based route system and other GIS layers. All the data files have been imported into or created within the TransCAD environment.

EU65 MetroCard Transactions

The EU65 MetroCard transaction file contains one record for each entry swipe or dip. It is harvested by NYCT from their AFC system and stored on a mainframe computer. Entry transactions are recorded by NYCT's subway turnstiles and bus fare boxes. Entry transactions are also obtained from the Port Authority of New York and New Jersey's JFK AirTrain and PATH Train (World Trade Center station), the Roosevelt Island Tramway, and a variety of buses in New York City operated by either sister MTA agencies or franchised private companies (now mostly MTA Bus). Table 1 shows the breakdown by transit mode of EU65 transactions for Tuesday, April 27, 2004. The EU65 file is by far this project's largest dataset; fourteen days of data (3am–3am) comprise 87,709,935 records, which are about 7GB in size.

Table 1: EU65 Transactions by Transit Mode for April 27 and 28

Mode	4/27 Transactions	4/27 Percentage	4/28 Transactions	4/28 Percentage
Bus (B)	2,690,147	36.27%	2,713,736	36.30%
SI Railway (I)	10,874	0.15%	10,730	0.14%
JFK AirTrain (J)	6,406	0.09%	6,559	0.09%
PATH Train (P)	1,731	0.02%	1,762	0.02%
Subway (S)	4,704,291	63.43%	4,740,355	63.41%
RI Tramway (T)	3,089	0.04%	3,132	0.04%
Total	**7,416,538**	**100.00%**	**7,476,274**	**100.00%**

Caliper suggested that the dataset be provided in the raw binary format, because it is more compact and can ensure a consistent format for later transfers. After several tries with a preliminary October 2003 dataset, we established that these files could be downloaded from the mainframe to an OMB PC using their EXTRA! software in binary mode, placed on an external hard drive, shipped to Caliper and converted to a TransCAD binary file. The EU65 data was grouped into six files: two each for bus, subway and student transactions. Breaking the datasets in half by week avoided the 4GB transfer limit with EXTRA!

Caliper developed in C, a standalone Windows utility program to convert the files efficiently from the mainframe COBOL format with EBCDIC strings to TransCAD's binary format with ASCII strings, discarding irrelevant data fields and converting HEET booth assignments based on an exception file. Any unreasonably large fare deduction (in excess of $10) is replaced with a missing value. The utility has been integrated into the Administrative Toolbox for the project that is described in more detail later in this report (page 55). Denied entry and first use records are discarded, since they do not represent actual trips. A single table is produced, sorted by MetroCard serial and transaction time, containing all the records for 15 days (midnight–midnight), from which daily extracts are created. The table has almost 95,000,000 records and requires twelve hours to create on the PC we used. The data dictionary for the TransCAD EU65 file is provided in Table 2. The data types are: C (Character String), I (4-byte Integer), S (2-byte Integer). Figure 1 depicts the processing of raw EU65 data.

From our examinations of both the preliminary October data and the April datasets, the EU65 data is of very high quality, with little cleanup required. A very small number of records had an unreasonable fare deduction in excess of $10 (35 records) or an invalid class (21 records not matching the table of MetroCard classes provided).

Table 2: EU65 Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Record ID
Serial	I	4	MetroCard Fare Card Serial Number
Date	I	4	Date of Transaction (Use)(YYYYMMDD)
Time	S	2	Time of Transaction (Use)(HHMM)
Type	C	3	Transaction Type
Class	C	3	MetroCard Class (Fare Media)
SCP	C	6	Station Controller Position ID (Address of End User Device)
Booth	C	5	Booth Number/Authority Code
Unit ID	C	4	Station Controller Computer/Depot
Value Deducted	S	2	Actual Value Deducted from Fare Card (cents)
Point of Entry	S	2	Point of Entry
Remaining Value	S	2	Remaining Value after last change (cents)
Authority	C	2	Authority Code
Transfer Valid	C	1	Entitled to a transfer or not
Trip Count	S	1	Trip Count
Transfer Count	S	1	Number of transfers allowed (0/1 unless group)
Denial Flag	C	1	Denial Flag
Borough	C	1	Borough Code
Mode	C	1	Mode (Bus, Subway, etc.)
Station	C	18	Station Name
Longitude	I	4	Fixed booth – Geographic location
Latitude	I	4	Fixed booth – Geographic location
When	I	4	Transaction time (MMDDHHMM)
Bus#	I	4	Bus Number (From SCP)

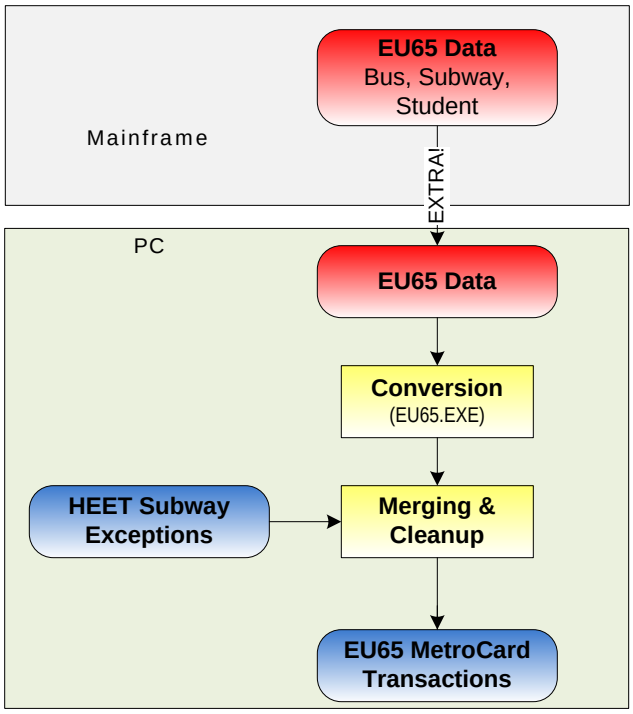

Figure 1: EU65 Data Flow

Bus Trip Table

The AFC bus trip table logs information from actual bus movements. One or more records should be present for each bus trip. Certain bus trips may be broken into multiple records due to various events, which include sign changes made by the driver, fixed times during the day (midnight, 6am, 9am, 4pm and 7pm), crew changes, etc. Occasionally, a record may represent more than one bus trip if an event failed to be logged correctly. We also observed a small number of corrupt records. As provided, the data is difficult to use, since we need a file with exactly one record per trip, so that we can easily and accurately identify the trip for each bus boarding. We created two GISDK scripts, now part of the Administrative Toolbox. The first converts the data into TransCAD's binary format and the second cleans it with the objective that the resulting table has one record per actual bus trip.

The AFC bus trip tables were received in a text format. There was one file for each bus operator: NYCT, Atlantic Express, Command Bus, Green Bus, Metro-North Hudson Rail Link, Jamaica Bus, Liberty Line, Long Island Bus, New York Bus, Queens Surface Transportation, and Triboro Coach. The file format is fixed-format ASCII (FFA) once the report heading at the beginning is removed.

The first GISDK script processes the raw files and converts them to a single binary file. The script concatenates all the input files into one master FFA file and opens it using a data dictionary that we created. This trip table is saved as a TransCAD binary file, keeping only the useful fields. The starting and ending dates and times are each converted into a single numeric field to facilitate matching with the EU65 transactions. Records covering times

outside of the study period are discarded. Depot and carrier information is added to the trip table, by a join with the bus depot database, matching PROBE_ID to UnitID. We dropped the 1145 records for the following depots, which either are closed or provide operator training: Hudson Pier (R493), LP-Depot (R501), Woodside Training (R503) and MDC Training (R508). Finally, we augmented the trip table with route information using the Transfer Table Master (tatmaster) files, provide by NYCT for all the bus operators (see Table 24). Tatmaster contain a list of sign codes, along with the sign text and information on the route pattern, including the loc code found in the EU65 file. Each operator is processed separately, since sign codes are only unique for a particular operator.

The raw bus trip table has one record per event; events can include the start of a run, a destination sign code change, a driver relief, or a time period transition. Table 3 lists the observed event codes along with their frequencies in the table we processed. Theoretically, there should have been one or more records per trip that could be easily combined to meet our needs. Unfortunately, this was not the case in practice. A few records appeared to be missing or corrupt, and others were invalid and represented multiple trips because some drivers failed to either sign-in or update their destination signs. The latter presented a significant obstacle.

Table 3: Bus Trip Event Codes

Event Code	Description	Observations
0	New Route/Run	188,514
1	Peak Period Transition	135,958
2	Time Period Audits	137,583
9	Bypass Start	782
10	Bypass End	479
11	TPU degraded mode start	10
12	TPU degraded mode end	43
13	New Preset	705,943
14	Midnight	14,128
15	Overflow	328
33	[Invalid]	1

Our first approach to this problem was to attempt to match the bus trips exactly to the schedule. Unfortunately, we only were able to match about 60% of the NYCT records using an automated approach, due to the variation between the schedule and the actual trips, and the split trip records and errors in the actual trip dataset. Our matching rate was too low to allow us to fill-in the missing trip information reliably.

Instead, a second GISDK script uses a simpler approach that first cleans the bus trips by combining records for a single trip where likely and then removes some obviously invalid records. This method works well in many cases, but not all. We postpone the matching with the schedule to a later stage and relax the process to only look for a similar trip in the schedule.

Our script starts with a conservative combination rule. The records are first ordered by PROBE_ID, FAREBOX_NUM, BUS_NUM, BLOCK_NUM and the starting time. Records are combined if they 1) Share the same values for the first four sort fields and for DRIVER_NUM and Sign, 2) Have no gap between their ending and starting times, 3) Have consecutive SEQUENCE_NUMs and 4) The later record does not start a new route/run. This rule fails to combine some records that represent the same trip.

Next we apply some heuristic rules to combine records further for NYCT trips. Some rules handle changes in the sign code to *Next Bus Please* (sign code 11) or *Not in Service* (sign code 12), when the bus is full. Another rule handles driver relief changes in the middle of a trip. The last rule combines the two halves of the Q48 route that have separate sign codes.

The final part of the script, cleans ups some NYCT sign codes to match the usage in the schedule. It also removes trips that are either shorter than five minutes, with an *Out of Service* sign code, or with obviously corrupted values.

Event code 15 is believed to indicate an overflow condition and probable loss of data, due to insufficient polling of the farebox. We suspect that trips and EU65 boardings recorded prior to the overflow event are available, but the subsequent trips and boardings are not included in the input datasets. It is doubtful that anything can be done for these rare events; there were 328 occurrences during the study period.

The resulting Bus Trip file has 915,658 records for the study period; see Table 4 for its data dictionary. Figure 2 depicts the processing of the AFC bus trip table.

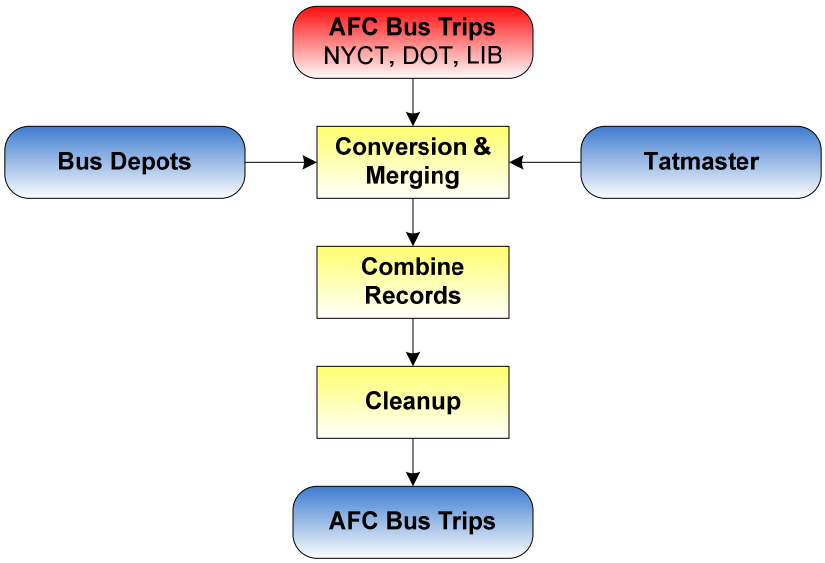

Figure 2: Bus Trips Data Flow

Table 4: Bus Trip Data Dictionary

Field	Type	Length	Description
Raw_ID	I	4	Unique Record ID
Key	S	28	Sort Order Key
Start	C	9	Start Time (MDDHHMMSS
End	C	9	End Time (MDDHHMMSS)
STime	I	4	Start Time (seconds)
ETime	I	4	End Time (seconds)
Duration	I	4	Trip Duration (seconds)
BUS_NUM	I	4	Bus Number
FAREBOX_NUM	I	4	Fare Box Number
Sign	I	4	Destination Sign Code
BLOCK_NUM	I	4	Block Number
RUN	I	4	Run
SEQUENCE_NUM	I	4	Sequence Number
BUS_ASGNED_DEP_CD	S	2	Bus Assigned Depot Code
PROBE_ID	C	4	Depot Unit ID
SCP	C	6	Station Controller Position ID
EVENT_CD	S	2	Event Code
DRIVER_NUM	I	4	Driver Number
DRVR_ASGNED_DEP_CD	S	2	Driver Assigned Depot Code
ZONE_CD	S	2	Zone Code
DIRECTION	S	2	Inbound/Outbound
PEAK_OFFPEAK	S	2	Peak/Offpeak
PRESET_NUM	S	S	Preset Number
TRIP_NUM	S	4	Trip Number
OWNING_AUTH	S	2	Owning Authority
PROBED_BORO	S	2	Probed Borough
PROBED_DEPOT	S	2	Probed Depot
Depot	C	18	Bus Depot Name
Carrier	C	3	Carrier Code
STIF Code	C	2	NYCT Depot Code used in STIF
Route	C	4	Route Name
Destination Sign Code Message	C	67	Destination Sign Code Message
Dir	C	1	Direction
Loc Code	S	2	Location Code
Res Group Loc Code	S	2	Res Group Location Code
Riders	I	4	Total Ridership

NYCT Bus Schedules

While the AFC system is common across all the bus and train operators, each operator uses its own scheduling system. For NYCT buses, the schedules were provided in the Surface Timetable Interchange Format (STIF). For each route, there are up to four separate schedule files, corresponding to Weekdays (schools open), Weekdays (schools closed), Saturday and Sunday. The schools-closed schedules did not apply to our study period and were ignored. Each file details the location of stops, the list of trips and the sequence of stops which

comprise a trip along the route. STIF is provided in service schedule order, which is distinct from crew order or equipment (or vehicle) order.

Caliper collated by borough all the STIF files that were received during 2004, keeping only the latest file for each route-service day combination. We applied a variety of checks to ensure that we had a complete set of schedules and did not have corrupted files. In a few instances, NYCT provided some replacement files to complete our set.

A GISDK script was developed to convert the STIF files into two TransCAD binary tables: the scheduled events and the stop locations. The procedure parses and expands all the raw STIF files, creating a table with one record per schedule event. Duplicate records, from trips occurring in more than one STIF file, were dropped. The resulting events file has over five million events that occur at more than 20,000 unique route-stops, corresponding to fewer physical locations. A separate TransCAD table of these route-stops is created by the procedure.

STIF is provided in service schedule order, but we need it in equipment order, to facilitate the location of bus dips. Deep in the documentation for STIF in section XII is a recipe for how to reorder the records into vehicle schedule order using Relief Route/Run/Time and Next Route/Run codes. We implemented this algorithm as part of our GISDK script. Initially, we encountered some errors in the relief route code field that were corrected by NYCT.

Unlike the sequence of trips forming an operator's run, we found no simple terminology for the sequence of trips made by a bus during a day. The STIF documentation refers to a segment of a schedule and cautions that subsequent segments during a day for a bus cannot be related. For lack of a better term, we decided to call the sequence of trips made by a bus a *job*.

Prior to 2005, STIF did not contain any geographic information for the stops. As part of the transition to the HASTUS scheduling system, DOT box numbers (unique identifiers) and 1927 state plane coordinates (zone 3101, NYS East, feet) were added for many of the stops. Unfortunately, the STIF Stop identifiers were changed between versions, so it was impossible to have a direct correspondence between the two versions.

We assigned DOT box numbers and coordinates to all of the 2004 stops. This messy and time consuming process included automatically matching as many of the 2004 routes to the 2005 routes as possible. Where possible, we used the file of DOT box locations to provide more accurate coordinates. The remaining stop locations were tediously located by geocoding the stop locations (street and cross street), cross-referencing with the DOT box file and route system where possible, with numerous manual checks. We had to create some new DOT box identifiers, for stops that were missing. The result is that we have unique geographic identifiers for each physical stop, often with an accurate coordinate. Those stops with less accurate coordinates are still in the general vicinity of the actual location and their accuracy should be no worse than that caused by the six minute rounding of transaction times.

Our script creates a list of distinct routes patterns from STIF (see Table 7). Ideally, these would all be present in the TransCAD route system and would match exactly, including all the stops. Unfortunately, this is not currently the case for bus routes, although we were able to achieve it for the subway routes. For buses, the stop locations are not accurate enough to automatically define additional routes and add missing stops. For this project, we were able to identify the equivalent route for most patterns, when it was in the route system. We automatically matched the sequence of stops where possible and the manually checked and corrected discrepancies. This yielded a good correspondence, but there are still stops present only in a STIF pattern and other stops present only in the route system.

The data dictionaries for the Bus Schedule Event and STIF Route Stop tables are provided in Tables 5 and 6.

Table 5: STIF Schedule Event Data Dictionary

Field	Type	Length	Description
Depot	C	2	Primary bus depot
Depot Name	C	24	Primary bus depot name
Borough	C	2	Operating/Owning Borough
Route	C	5	Route Identifier
Service	C	2	Service Period
Route Description	C	24	Brief Description of Route
Schedule	C	10	Schedule Number
Organization	C	2	Operating Organization
Origin	C	4	Origin Location
Start Time	I	4	Starting Time
Direction	C	1	Direction of Service
Trip Type	C	2	Type of Trip
Destination	C	4	Destination Location
End Time	I	4	Ending Time
Run	C	3	Primary Run Number
Path	C	5	Pattern Code
Primary Route	C	5	Primary Route
Relief Run	C	3	Mid-trip Operator Relief Run Number
Relief Route	C	5	Mid-trip Operator Relief Route
Relief Time	I	4	Time of Mid-trip Operator Relief
Relief Location	C	4	Location of Mid-trip Operator Relief
Bus Type	C	1	Bus Type
Sign	C	4	Sign Code
Sign Route	C	5	Sign Route Name
Sign Text	C	58	Sign Text
First Trip	C	1	Whether it is the first trip
Last Trip	C	1	Whether it is the last trip
Primary Relief	C	1	Whether the relief operator will clear
Next Run	C	3	Run number of next operator to drive bus
Next Route	C	5	Route of next operator to drive this bus
Event Location	C	11	Unique Stop ID
Event Time	I	4	Event Time

Event Type	C	1	Type of Event
Stop	C	1	Whether this is a stop
Timepoint	C	1	Whether this is a timepoint
Route_Name	C	10	Schedule Route Name
Route_ID	I	4	TransCAD Route Identifier
Trip	I	4	Trip Identifier
Seq	I	4	Stop Sequence within Trip
Stop_ID	I	4	TransCAD Stop Identifier
Box	I	4	DOT Box Number
Milepost	R	8	TransCAD Route Milepost
NewTime	I	4	Time interpolated by dist between timepoints
Longitude	I	4	Geographic Coordinate - Caliper
Latitude	I	4	Geographic Coordinate - Caliper
Distribution	R	4	Used for Sampling

Table 6: STIF Route Stop Data Dictionary

Field	Type	Length	Description
Stop	C	11	Unique Stop ID
Location Description	C	22	Stop Location Description
Intersection	C	22	Stop Intersection Description
Borough	C	2	Borough
Timepoint	C	5	Whether the stop is a timepoint
Location Type	C	1	Depot or Normal Stop
X	I	4	Geographic Coordinate
Y	I	4	Geographic Coordinate
DOT Box	I	4	DOT Box from STIF
Box	I	4	DOT Box assign by Caliper

Table 7: STIF Route Patterns Data Dictionary

Field	Type	Length	Description
Route_Name	C	10	STIF Route Name
Sign	C	4	Sign Code
NStops	I	4	Number of Scheduled Stops
Trip	I	4	Trip Identifier (from full schedule)
Duration	I	4	Duration of scheduled trip (min)
Matching_Route	C	32	TransCAD Route Name
Matching_NStops	I	4	Number of Route Stops
Matching_Score	I	4	Score from Matching Algorithm
NStops_Matched	I	4	Number of matched stops

We created subsets of the STIF schedule for each service period active during the study period: STIF1.Bin for weekdays, STIF2.Bin for Saturdays and STIF3.Bin for Sundays. We also create tables of scheduled trips: Trips1.Bin, Trips2.Bin and Trips3.Bin (see Table 8).

Table 8: STIF Trips Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Identifier
SWork	C	8	Depot + Service + Unique Identifier
Start	I	4	Start of Trip (HHMM)
End	I	4	End of Trip (HHMM)
Route	C	5	NYCT Route
Run	S	2	Primary Run Number
Sign	S	2	Destination Sign Code
Path	C	4	Pattern
Trip Type	S	2	Trip Type
			1 – Normal Revenue Trip
			2 – Pull-out from depot
			3 – Pull-in to depot
			4 – Deadhead
			5 – Trip transition
			11 – School service
			12 – Limited service
			13 – Express service
Direction	C	1	Direction of Service
Record	I	4	STIF Record Identifier
Event	C	13	Special events, used for reordering
SKey	C	14	Prev Sign + ":" + Sign:Path
Smin	I	4	Start of Trip (min)
Emin	I	4	End of Trip (min)
Duration	I	4	Duration of Trip (min)
DKey	C	14	Depot + Run + ":" + Sign + Path

Long Island Bus Schedules

The Long Island Bus (LIB) is a sister MTA agency that operates the Nassau County bus system on Long Island. Some of their routes service portions of Queens and they use the MetroCard system, so it was desirable to include these services in the project. LIB uses the Trapeze Scheduling Software. We received twelve fixed-format ASCII files that detailed their routes, trips, stops and schedule. These included accurate geographic representations of the stops, so we were able to add all the LIB route patterns to the TransCAD route system and have a near perfect correspondence between the schedule and route system.

We applied data dictionaries to each of the LIB files, so that they could be used with TransCAD. We manually processed all the information, so that we had a schedule event file. We also added their stop locations to our DOT box file. Where the LIB routes overlap with other operators, the same physical stop may be used for each.

We created subsets of the LIB schedule for each service period active during the study period (see Table 9): LIB1.Bin for weekdays, LIB2.Bin for Saturdays and LIB3.Bin for Sundays.

We also create tables of scheduled trips: Trips1.Bin, Trips2.Bin and Trips3.Bin (see Table 10).

Table 9: LIB Schedule Event Data Dictionary

Field	Type	Length	Description
Depot	C	2	Primary bus depot
Depot Name	C	24	Primary bus depot name
Borough	C	2	Operating/Owning Borough
Route	C	5	Route Identifier
Service	C	1	Service Period
Route Description	C	37	Brief Description of Route
Schedule	C	3	Schedule Number
Origin	I	4	Origin Location
Start Time	I	4	Starting Time
Direction	C	3	Direction of Service (N, S, E, W, CW, CCW)
Trip Type	C	1	Type of Trip
Destination	I	4	Destination Location
End Time	I	4	Ending Time
Run	I	4	Primary Run Number
Path	C	4	Pattern Code
Sign	C	4	Sign Code
Sign Route	C	5	Sign Route Name
Sign Text	C	58	Sign Text
Last Trip	C	1	Whether it is the last trip
Event Location	I	4	Unique Stop ID
ETime	I	4	Event Time
Event Type	C	1	Type of Event
Timepoint	C	1	Whether this is a timepoint
Trip	I	4	Trip Identifier
Seq	S	2	Stop Sequence within Trip
Distance	R	8	Distance (m) from start of segment
Location	C	30	Stop Location
Intersection	C	25	Intersecting Street
Key	C	10	Sign + ":" + Path
Box	I	4	DOT Box Number
Longitude	I	4	Geographic Coordinate - Caliper
Latitude	I	4	Geographic Coordinate - Caliper
Stop	C	1	"S" – Always a stop
Event Time	I	4	Event Time
Stop_ID	I	4	TransCAD Stop Identifier
Route_ID	I	4	TransCAD Route Identifier
Distribution	R	4	Used for Sampling

Table 10: LIB Trips Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Identifier
Route	C	5	LIB Route
Run	S	2	Primary Run Number
Sign	S	2	Destination Sign Code
Path	C	4	Pattern
Trip Type	C	1	Trip Type
			1 – Normal Revenue Trip
Direction	C	1	Direction of Service
Trip	I	4	Trip Identifier
Smin	I	4	Start of Trip (min)
Emin	I	4	End of Trip (min)
Duration	I	4	Duration of Trip (min)
Route_ID	I	4	TransCAD Route Identifier

The quality of LIB's schedule information made it the most straightforward of any bus operator to process and handle.

Metro-North Hudson Rail-Link Bus Schedules

Metro-North's Hudson Rail Link consists of a few connecting bus routes to their Hudson Commuter Train line at the Riverdale and Spuyten Duyvil stations. Their schedules are available in tabular format on MTA's website, along with maps. We captured the information and converted into TransCAD. While we were able to add the Hudson Rail-Link routes, the schedule and sign code information was too patchy to use for locating dips. Given that there are only 300 riders per day and it serves primarily as a feeder service to MetroNorth, we did not locate these transactions.

DOT Bus Schedules

There were seven private bus companies that operated routes franchised by New York City's Department of Transportation, and, after a long process, have been taken over by the MTA and are now operated as MTA Bus. Unfortunately, during our study period, the relations with some of the companies were rather poor, and they were not forthcoming with information. We didn't receive much additional information once MTA took control. Depending on the company, we received differing levels of details, provided as Excel spreadsheets and/or captures from their websites. Not all the schedules were of the correct vintage for our study period and none of them included vehicle schedules or the level of detail found in STIF or LIB's files.

We undertook the tedious task of combining all the available information, including the tatmaster table and TransCAD route system and manually produced a schedule file. We also added missing stops with our best available location to the DOT box file. The result allows the DOT transactions to be processed, but with a much lower level of confidence than for the

other buses. It was not anticipated that we would have to support multiple schedule formats for the different bus operators in New York City, other than NYCT.

Caliper created subsets of the DOT schedule for each service period active during the study period (see Table 11): DOT1.Bin for weekdays, DOT2.Bin for Saturdays and DOT3.Bin for Sundays. We also create tables of scheduled trips: DOT_Trips1.Bin, DOT_Trips2.Bin and DOT_Trips3.Bin (see Table 12).

Table 11: DOT Schedule Event Data Dictionary

Field	Type	Length	Description
Trip	I	4	Trip Identifier
Seq	I	4	Stop Sequence within Trip
Location	C	25	Stop Location
Intersection	C	30	Intersecting Street
Time	I	4	Event Time (HHMM)
Timepoint	C	1	Whether this is a timepoint
Trip Type	C	1	Type of Trip
Event Time	I	4	Event Time
Route_ID	I	4	TransCAD Route Identifier
Stop_ID	I	4	TransCAD Stop Identifier
Box	I	4	DOT Box Number
Longitude	I	4	Geographic Coordinate - Caliper
Latitude	I	4	Geographic Coordinate - Caliper
Stop	C	1	"S" – Always a stop
Pattern	C	10	Schedule Pattern Code
Distribution	R	4	Used for Sampling

Table 12: DOT Trips Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Identifier
Trip	I	4	Trip Identifier
Route_ID	I	4	TransCAD Route Identifier
Pattern	I	4	Pattern Identifier
Smin	I	4	Start of Trip (min)
Emin	I	4	End of Trip (min)
Duration	I	4	Duration of Trip (min)
Path	C	4	Pattern
Sign	I	4	Destination Sign Code

Atlantic Express Bus Schedules

Atlantic Express is a private company which still operates a few express bus routes between Manhattan and Staten Island via New Jersey. They provided their schedule in Excel Format. Given their limited service with very simple routes, we were able to add accurate information

to our DOT schedule files, tatmaster table and route system. We have good confidence in the location of their transactions.

Subway Schedules

The NYCT Subways provide schedules in Rapid Timetable Interchange Format (RTIF), which is similar to the STIF files. For each line, there are up to three separate files, corresponding to Weekdays, Saturday and Sunday. Each file details the stations, the list of trips and the sequence of stations that comprise a trip along the route.

Caliper developed a GISDK script that is part of the administrative toolbox, to convert the RTIF files into a TransCAD binary table. The procedure opened each schedule file in turn and expanded the data into a single table with a total 595,876 records, each corresponding to a schedule event that occurs at 2,549 unique route-stops but at fewer physical locations. A separate TransCAD table of these route-stops was created by the procedure. The procedure also created a table of unique route patterns. These were automatically converted into routes with stops in the TransCAD route system. We also converted the subway schedules into required TransCAD format, for use with our schedule-based shortest path (SSP), which is discussed later in this report.

The data dictionaries for the RTIF Schedule Event and Route Stop tables are provided in Tables 13 and 14.

Table 13: RTIF Event Data Dictionary

Field	Type	Length	Description
Line	C	2	Subway Line
Service	C	1	Service Period
Schedule	C	4	Schedule Number
Supplement	S	2	Supplement Number
Origin	C	3	Origin Location
Start Time	I	4	Starting Time
Direction	C	1	Direction of Service
Trip Type	C	2	Type of Trip
Destination	C	3	Destination Location
End Time	I	4	Ending Location
Run	C	3	Run Number
Work Pgm	C	2	Work Program ID
Path	C	7	Pattern Code
Trip Line	C	2	Trip Line
Equipment	C	3	Equipment Type
Cars	S	2	Number of Cars Required
Relief Pgm	C	1	Relief Program ID
Relief Run	C	3	Relief Run Number
Relief Loc	C	3	Relief Location
Relief Time	I	4	Relief Departure Time
Next Trip	C	1	Next Trip Type

Next Time	I	4	Next Trip Time
Crewing	C	2	Crewing
Event Location	C	7	Unique Station ID
Event Track	C	2	Track
Event Time	I	4	Event Time
Event Type	C	1	Type of Event
Stop	C	1	Whether this is a stop
Timepoint	C	1	Whether this is a timepoint

Table 14: RTIF Route Stop Event Data Dictionary

Field	Type	Length	Description
Station	C	7	Unique Station ID
Abbrev	C	8	Station Abbreviation
Name	C	32	Station Name
X	I	4	X Coordinate
Y	I	4	Y Coordinate

Staten Island Railway Schedules

The Staten Island Railway (SIR) operates a single line from Ferry Terminal at St. George to the opposite end of the Island. We received their schedule as an Excel spreadsheet with sheets for weekday and weekend service periods. We converted the SIR schedule into the TransCAD's SSP format.

JFK AirTrain Schedules

The Port Authority of New York and New Jersey's JFK AirTrain provides service from the Howard Beach and Jamaica stations to JFK Airport. Their website provides information on the routes and the approximate headways during different time periods. This is insufficient for application of TransCAD's SSP procedure. We are unable to tell which terminal a passenger used and whether a passenger arrived at and departed from the airport, so we did not attempt to determine the AirTrain trips.

PATH Schedules

The Port Authority also operates the PATH train underneath the Hudson River. Their World Trade Center station now accepts MetroCard, so their transactions are included in the EU65 file. It was straightforward to convert their schedules from their website into TransCAD's SSP format.

We originally considered adding PATH trip legs for WTC boardings, but had no information regarding where the alightings occurred. Instead, we used handled these trips in the same fashion as the other commuter rail and bus transfers.

Roosevelt Island Tramway Schedules

The Roosevelt Island Tramway operates between 59th Street & Second Avenue in Manhattan and Roosevelt Island. Their website provides headways for different time periods and we converted this into a schedule in TransCAD's SSP format.

Staten Island Ferry Schedules

NYC operates the Staten Island Ferry between Whitehall and St. George. It effectively connects the subway system and the SIR, so it is important to handle flows between the two boroughs. From information on its website, we created a schedule in TransCAD's SSP format.

Transit GIS Route System

In addition to the transactional and schedule data, the other key database for this project is the TransCAD route system that provides a detailed geographic representation of the transit routes in NYC. Prior to the start of this project, OP was using an older route system based on the less spatially precise 1994 LION files for modeling and that only included AM Peak routes. OP desired to migrate to a route system based on the more accurate NYCMap centerlines and a more complete route system was needed for this project. To that end, Caliper collaborated with OP to create a new enhanced route system; we supported the goals of switching to a route system to the new base map for the improved geographic accuracy and the use of physical stops, since they are important to the quality of the project's outputs. Physical stops are an optional enhancement to a route system that allow many routes to share a common point for a stop, guaranteeing that such route stops are located at the same geographic coordinate. Creating an enhanced route system was a big undertaking, not anticipated in the work plan or budget for the project.

A preliminary route system was received from OP with street centerlines and bus routes. The centerlines were from NYCMap, except outside of the city. Caliper wrote an automated procedure to locate and add physical and route stops using the NYC DOT spreadsheets and add their Box number as a unique reference. OP then added the subway linework using NYCMap for above ground features and another accurate source for below ground. We then transferred the subway routes and stations from their old route system.

There were several iterations of route system development during which Caliper swapped improvements in the route system with OP. OP spent numerous hours checking routes and improving the location and completeness of route stops.

Using the complete subway schedules that we extracted from RTIF, we added all the missing subway patterns to the route system, including weekend and single trip variants. For the existing subway routes, we verified that all the stops matched the schedule. We changed all the subway route names to match RTIF and they now include RTIF's pattern code (e.g. 7..N01R).

Caliper also added to the route system, linework, routes and stops for the other transit operators in NYC, so that the route system includes the ferries (Staten Island, Governor's Island and Liberty/Ellis Islands), the Roosevelt Island Tramway, the JFK AirTrain, and PATH. Except for the small ferries and the AirTrain, there is now a complete and accurate representation with schedules for all the non-bus transit links making it possible to use TransCAD's SSP procedure to choose a route for individual trips. Linework and station nodes (within NYC) were also added for Metro-North and LIRR.

The NYCT bus schedules, extracted from STIF, were used to improve the route system and add location information to the schedules. Unfortunately, we did not get a perfect correspondence. OP worked hard to ensure that at least the major route patterns were in the route system. We were unable to automatically add the missing patterns, like we did for subway, due to the lack of accurate stop locations in the schedule and the multiple paths that could be used on the street network.

The DOT bus routes in the route system were in a much worse condition and while we spent hours trying to match and improve the routes and schedules, we could not achieve the same level of results as we did for the NYCT buses. Since the inception of MTA Bus, OP has continued to improve their route system, particularly for the old DOT routes. Unfortunately, due to the budgetary constraints for this project, we were unable to make use of this information,

For the Long Island Bus, we extracted linework from their schedule file and merged it into the NYCT line layer. We also extracted their physical stops and created a TransCAD route system covering the LIB service area (Queens, Nassau and Suffolk counties), using the merged line layer. We then merged this route system into our master route system. The quality of LIB's schedule files, allowed us to create the most complete and accurate route system for any major bus operator.

We also added the few routes for MetroNorth's Hudson Rail Link bus service, using definitions available on their website. We also added bus depot locations and connectors, so that complete trips can be mapped if desired.

OP decided to use 2000 Census Block Groups for their new zone layer for modeling. We applied an automated tool to create centroid connectors for the new zones. Typically this added at least two connectors for each zone. Several iterations with this tool were required before we had connectors for every zone in the city. We also added artificial zone centroids for commuter transit transfer flows from/to PATH, MetroNorth, LIRR and New Jersey Transit,

The resulting route system from all OP and our efforts is sufficient for the project and as a basis for OP's transportation planning. It is still uneven in its quality and a reasonable future goal would be for it to evolve to include all the bus route patterns with all the bus stop locations and accurate paths through the street network.

OP continues to refine and improve the quality of the bus routes. This continuing effort certainly is improving the route system. We had not anticipated the difficulty of making use of new versions of the route system. In order for our procedures to work, we need careful coupling between the schedules, bus trips and route system. This means that the stops and route identifiers must match and the sequence of stops used should be in increasing order. This requires manual effort to insure the consistency, so we were only able to make use of newer versions of the route system a few times. It proved tricky to merge our improvements required for the project with the parallel improvements from OP. This endeavor was not in the work plan, so we were only able to perform it a few times.

The data dictionaries for the various layers associated with the route system are documented in Tables 15-19.

Table 15: Routes Layer Data Dictionary

Field	Type	Length	Description
Route_ID	I	4	Unique Route Pattern Identifier
Route_Name	C	32	Unique Route Pattern Name
Route	C	14	Route Name
Dir	C	3	Route Direction
Path	C	7	Path Code
Destination	C	37	Route Destination
Length	R	8	Route Length (mi)
Operator	C	20	Transit Operator
Carrier	I	4	Bus Carrier Code
Mode No	I	4	Transit Mode Code
			1 = Local Bus
			2 = Limited Bus
			3 = Express Bus
			5 = Subway
			6 = Aerial Tramway
			7 - Ferry
Mode	C	11	Transit Mode
EU65 Mode	C	1	Transit Mode EU65 Code
			B = Local & Limited Bus
			E = Express Bus
			F = Ferry
			I = Staten Island Railway
			J = JFK AirTrain
			P = PATH
			S = Subway
			T = Aerial Tramway
Veh Cap'y	R	8	Vehicle Capacity
AM Headway	R	8	AM Peak Headway (min) - Old
AM Combined Headway	R	8	AM Peak Combined Headway (min) - Old
AM Veh/Hour	R	8	AM Peak Vehicles/hour - Old
Midday Headway	R	8	Midday Headway (min) - Old

Midday Combined Hdwy	R	8	Midday Combined Headway (min) - Old
Midday Veh/Hour	R	8	Midday Vehicles/hour - Old
PM Headway	R	8	PM Peak Headway (min) - Old
PM Combined Headway	R	8	AP Peak Combined Headway (min) - Old
PM Veh/Hour	R	8	PM Peak Vehicles/hour - Old
ALPHA	R	8	Assignment crowding penalty
BETA	R	8	Assignment crowding penalty
AM Pk Pd	C	1	Indicates whether Route is active during AM Peak
LIB_Route	I	4	LIB Route ID
Division	I	4	Dummy field, required by SSP, not used
ModeID	I	4	Dummy field, required by SSP, not used
CIS	C	1	Route should be included in SSP Network
Pattern	C	10	Pattern Code
N_Trips	I	4	Number of Scheduled Trips
RF_Travel_Time	R	8	Average Travel Time from Schedules
RF_AM	C	1	AM Peak Period Route
RF_Headway	R	8	Average Headway from Schedules
New_Headway	R	8	Best Available Headway
New_Capacity	R	8	Best Available Capacity
Express	C	1	Non-Local Subway Route
			E = Express
			L = Limited
			SS = Shuttle

Table 16: Stops Layer Data Dictionary

Field	Type	Length	Description
Stop_ID	I	4	Unique Stop Identifier
Box	I	4	DOT Box Number/Station ID
Station/Location	C	50	Station Name/Street Intersection
Route_Name	C	32	Route Pattern Name
Route	C	14	Route Name
Dir	C	1	Route Direction
Path	C	7	Path Code
Mode	C	11	Transit Mode
Snap Node	I	4	Nearest Node - Old
Used	C	1	Used by scheduled trips
Sch_Seq	F	4	Sequence within Schedule Pattern
Pattern	C	10	Schedule Pattern
Check	C	1	To be deleted
NODEID	I	4	Nearest Node
REALSTOP	I	4	To be deleted
Cordon	C	5	Manhattan CBD Cordon Station
In/Out	C	1	Cordon Inbound/Outbound
Key	C	8	Cordon Key Field
Borough	S	1	Borough Code
			0 = New Jersey
			1 = Lower Manhattan CBD

			2 = The Bronx
			3 = Brooklyn
			4 = Queens
			5 = Staten Island
			6 = Remainder of Manhattan
			7 = Nassau County
			8 = Suffolk County
			9 = Westchester County
Peak	C	1	AM Peak Period Stop
TT	R	8	Travel Time (min)
ATT	R	8	? Travel Time (min)
Speed	R	8	Vehicle Speed (mph)
Zero	C	1	
X	I	4	Unique ID

Table 17: Physical Layer Data Dictionary

Field	Type	Length	Description
Stop_ID	I	4	Unique Physical Stop Identifier
Box	I	4	DOT Box Number/Station ID
Station/Location	C	50	Station Name/Street Intersection
Route(s)	C	19	Subway Route(s) using link
Mode	C	11	Transit Mode
LIB_Stop	I	4	LIB Stop Identifier

Table 18: Line Layer Data Dictionary

Field	Type	Length	Description
Type	S	2	Link Type
			1 = Street
			2 = Subway Track
			3 = Subway Transfer
			4 = Subway Entrance
			5 = Subway Mezzanine
			6 = Ferry
			7 = Ferry Entrance
			8 = Tramway
			9 = Tramway Entrance
			10 = Commuter Rail Track
			99 = Bus Depot Connector
Mode	S	2	Link Mode
			8 = Walk Link
			9 = Connector Link
Name	C		Street/Track Name
Route(s)	C	43	Subway Route(s) using link
IVTT	F	4	In-Vehicle Travel Time (min)
WalkTime	F	4	Walking Time (min)

Non-Transit Link	C	1	Link should be included in Walking Network
CIS	S	1	Link should be included in SSP Network
Overlap ID	I	4	To be deleted
N Intersections	S	2	To be deleted
Connector	I	4	Zone Number if Centroid Connector
Agency	C	20	External Commuter Station Operator
Ostation	C	30	External Commuter Station Name
Obox	I	4	External Commuter Station ID
Complex	I	4	External Commuter Nearest Subway ID
Station	C	30	External Commuter Nearest Subway Name

Table 19: Intersection Layer Data Dictionary

Field	Type	Length	Description
Station/Location	C	50	Station Name/Street Intersection
Div	C	11	Subway Division
Line	C	42	Subway Line
Route(s)	C	19	Subway Route(s) using link
Nearest Street Intersection	I	4	Nearest Street Intersection Node ID
NB_PStop	I	4	Northbound Physical Stop ID
SB_PStop	I	4	Southbound Physical Stop ID
Singleton	C	1	To be deleted
Street	C	1	Node connects to a street
TAZ	C	11	1990 TAZ - To be deleted
County	C	5	County FIPS Code
Tract	C	11	Census Tract Code - To be deleted
Centroid	I	4	Zone ID - Node is Centroid
Active	C	1	Stop is used by schedule

Station List

A master list of active subway stations with coordinates was created for use as a GIS reference layer and for certain steps in our location procedure. The few stations closed for construction were excluded to prevent possible use in passenger trips. The list includes SIR stations, SIF terminals and certain PATH stations.

Additional fields were added to the table, which include the subway complex ID for large stations, physical stop IDs from the new TransCAD route system, a list of nearby stations with estimate walking times for SSP queries, and a list of nearby zones for zone allocations.

The stations IDs are in the range 1000-1466 for NYCT, 1700-1721 for SIR and 1800-1899 for other agencies. This is outside the range of DOT Box numbers, which are all six digits.

Table 20 documents the Station List layer.

Table 20: Station List Data Dictionary

Field	Type	Length	Description
Box	I	4	Station ID
Node	I	4	Intersection Node ID
TAZ	C	11	1990 Zone
Station/Location	C	43	Station Name
Div	C	11	Subway Division
Line	C	42	Subway Line
Route(s)	C	19	Subway Route(s) using link
NB_PStop	I	4	Northbound Physical Stop ID
SB_PStop	I	4	Southbound Physical Stop ID
NB_PStop2	I	4	Alternate Northbound Physical Stop ID
SB_PStop2	I	4	Alternate Southbound Physical Stop ID
Mixed_Dir	C	1	Route N/S usage is inconsistent
NearN1	I	4	Nearby Station - NB Physical Stop ID
…	I	4	Nearby Station - NB Physical Stop ID
NearN5	I	4	Nearby Station - NB Physical Stop ID
NearNF	I	4	Nearby SIF - NB Physical Stop ID
NearS1	I	4	Nearby Station - SB Physical Stop ID
…	I	4	…
NearS5	I	4	Nearby Station - SB Physical Stop ID
NearSF	I	4	Nearby SIF - SB Physical Stop ID
Time1	F	4	Walking Time to Nearby Station (min)
…	F	4	…
Time5	F	4	Walking Time to Nearby Station (min)
TimeF	F	4	Walking Time to Nearby SIF (min)
Cnode	I	4	Station Complex ID
StartPeak	I	4	Start of AM Peak Hour (HMM)
EndPeak	I	4	End of AM Peak Hour (HMM)
Model	I	4	Old Model Node ID
Complex	I	4	Station/Complex ID
Borough	S	1	Borough Code
BG	C	12	2000 Block Group Code
Distribution	F	4	Sampling Distribution
NBG	S	1	# Allocation Block Groups
BG1	I	4	Nearby Block Group 1
…			…
BGn	I	4	Nearby Block Group n

Booths

A master list of subway booths, with coordinates and station IDs was also created. This was compiled from OP's older route system, an Excel Spreadsheet of fare control locations obtained from OP and the list of booths with TransCAD nodes obtained from OMB. This created a list of subway booths, with information on the station, the serving subway lines and coordinates that can be joined to the EU65 transactions, documented in Table 21.

Table 21: AFC Booths Data Dictionary

Field	Type	Length	Description
Lon	I	4	Geographic Location
Lat	I	4	Geographic Location
Booth	C	5	Booth Code
Unit ID	C	4	Unit ID
Borough	S	1	Borough Code
Mode	C	1	Transit Mode EU65 Code
Station	C	25	Station Name
Box	I	4	Station ID
Direction	C	1	Direction Restriction
NB_PStop	I	4	Northbound Physical Stop ID
SB_PStop	I	4	Southbound Physical Stop ID
Mixed Dir	C	5	Route N/S usage is inconsistent
Routes	C	30	Subway Route using link
Start Time	I	4	Start of AM Peak Hour (HMMSS)
End Time	I	4	End of AM Peak Hour (HMMSS)
Shift	S	2	Entry/Exit Register Count Hour Shift
TCNode	I	4	Old TransCAD Route System Node ID
HEET	C	1	HEET Turnstile
Closed	C	1	Station Closed
Location	C	48	Station Location
Cnode	I	4	Station Complex ID

Bus Depot List

We created a master list of bus depots with coordinates for use as a GIS reference layer. We combined the database of NYCT bus depots provided by OP with the list of all AFC bus depots from OMB into a master database of bus depots. Locations for the other depots were added using addresses obtained via a web search and the DOQ images as a reference, except that no accurate location was discovered for the Hudson Rail buses. Table 22 documents the Bus Depot table.

Table 22: Bus Depot Data Dictionary

Field	Type	Length	Description
Depot	C	18	Depot Name
Address	C	30	Street Address
City	C	30	Postal City
County	C	13	Borough/County Name
UnitID	C	4	Depot UnitID Code
Carrier	C	3	Carrier Number
Code	S	1	Depot ID
Abbrev	C	30	Depot Abbreviation
STIF Code	C	2	STIF Depot Code
STIF Name	C	18	STIF Depot Name
Box	I	4	DOT Box Number

DOT Box List

A master list of bus stop locations was compiled as part of our processing of the schedules and the improvements that were made to the route system. We started with NYC DOT's spreadsheet of locations and combined it with LIB's stop locations. We added additional records for bus stops not in the DOT list and for bus depots.

Each record contains the street and cross street for the stop. We added the nearest TransCAD intersection and a list of nearby zones for zone allocations.

The boxes are in the range 3-99 for depots, 100000-199999 for the Bronx, 200000-299999 for Staten Island, 300000-399999 for Brooklyn, 400000-499999 for Manhattan, 500000-599999 for Queens, 700000-799999 for LIB, and 900000-999999 for new stops added for the project. Table 23 documents the box table.

Notice that the station ids and box numbers do not overlap. For some parts of this project, we use a combined list. We created a merged table of ID and display name for the query software.

Table 23: Box Data Dictionary

Field	Type	Length	Description
Box	I	4	DOT Box Number
Location	C	40	Street
Intersection	C	40	Cross-street
Longitude	I	4	Geographic Location
Latitude	I	4	Geographic Location
Borough	C	2	Borough Abbreviation
Boro	S	1	Borough Code
StartPeak	S	2	To be deleted
BG	C	12	2000 Block Group
Route	C	10	
Node	I	4	Nearest Intersection Node
BG	C	12	2000 Block Group Code
NBG	S	1	# Allocation Block Groups
BG1	I	4	Nearby Block Group 1
…			…
BGn	I	4	Nearby Block Group n
TA_AM	I	4	NYCT AM
DOT_AM	I	4	DOT AM
LIB_AM	I	4	LIB AM

Tatmaster

As part of processing the bus schedules, we compiled a master table of destination sign codes, documented in Table 24. This combined tatmaster tables for NYCT and the other

buses into a single file with information on the carrier, depot, route, sign code and loc code. Additional key fields allow us to determine the route patterns for particular bus trips.

Table 24: Tatmaster Data Dictionary

Field	Type	Length	Description
Key	C	8	Lookup Key (Carrier:Sign Code)
Operator	C	20	Transit Operator
Carrier	S	1	Carrier Number
Route	C	6	Route
DEST SIGN CODE	S	2	Destination Sign Code
DESTINATION SIGN CODE MESSAGE	C	67	Destination Sign Text
DIR	C	4	Direction
DIV	C	1	Unused
DEPOT	C	64	Bus Depot
LOC CODE	S	2	Location Code
RES GROUP LOC CODE	S	2	Alternate Location Code

By combining information from the schedules, tatmaster and the route systems, a list of unique bus patterns was created (see Table 25).

Table 25: Pattern Data Dictionary

Field	Type	Length	Description
Loc	S	2	Location Code
Sign	S	2	Destination Sign Code
Direction	S	1	Direction Code (0/1)
Path	C	4	Pattern
Route_Name	C	10	Schedule Route Name
NRSName	C	15	
Exact	C	1	Exact Match to Route System (X/N)
RSName	C	32	TransCAD Route Name
Route_ID	I	4	TransCAD Route Identifier
Operator	C	20	Transit Operator
Key	C	5	
KeyS	C	9	
Carrier	S	2	Carrier Code
PKey	C	13	Carrier + ":" + Loc + ":" + Sign
Mode	C	1	L – Limited, E - Express
N428	I	4	Number of 4/28 Boardings
NBT	I	4	Number of Bus Trips
O	C	11	Origin
D	C	11	Location

GIS Reference Layers

Caliper created and compiled a number of reference layers for use with this project.

There are three government sources of GIS data for New York City: NYCMap from NYC Department of Information Technology & Telecommunications (DoITT), DCLION from NYC Department of City Planning (DCP) and TIGER from the US Census Bureau. The latter provides reasonable information nationwide, but is unfortunately spatially less accurate.

DCP provides on its website, street centerlines from the DCLION database as part of the BYTES of the BIG APPLE. These files originated as an early version of TIGER, but have been locally maintained and significantly improved using color Digital Orthophotography (DOQ). Unfortunately, they have not been completely realigned yet, so they do not entirely match the NYCMap centerlines. The latter has wonderful spatial accuracy, but no address ranges for geocoding. NYCMap and the NYC DOQs are licensed by DoITT to OP, which provided Caliper with a sublicense for this project. The NYCMap street and rail centerlines are used by the new OP route system described in this report.

We converted the MapInfo version of the DCLION files for each borough and merged them into a single city wide street layer for use in geocoding intersections and addresses. Intersections were used to geocode physical bus stops and the addresses for the Seniors' MetroCard addresses. The alternate street names, including common misspellings, were extracted so that provisions can be made to more accurately geocode addresses for this project.

We also used the DCLION files to create a 2000 Census Block layer with geography consistent with the streets. Caliper's standard demographics from the 2000 Census were attached to the layer. This layer was also used to create 2000 Census Tract, Census Block Group and Borough layers.

Once OP decided to use Block Groups for their new zone layer, Caliper combined the LION based layer for NYC with TIGER Block Groups for parts of Nassau, Suffolk and Westchester counties. We added artificial zones for commuter flows on MetroNorth, LIRR, PATH and New Jersey Transit (NJT). These zones correspond to the stations in NYC that the commuter enters/leaves the NYC transit system; e.g., MetroNorth riders using Grand Central. We attached to this layer, demographic fields containing population aged 15+ and employment from the 2000 CTPP dataset. We also added land use information that OP aggregated from DCP's tax parcel data that details the amount of different types of space (residential, office, retail, etc.) Table 26 documents the Zone layer.

Table 26: TAZ (Block Group) Data Dictionary

Field	Type	Length	Description
BG	C	12	2000 Block Group
Borough	C	1	Borough Code
Boro	C	1	To be deleted
County	C	5	County FIPS Code
Zone	I	4	Zone ID (BG without State FIPS)
Station	C	40	Station Name
TAZ	C	20	
Box	I	4	Station ID/DOT Box Number
LIB	C	1	Serviced by LIB
Pop15+	S	2	2000 Population Aged 15+
Employees	I	4	2000 Employees
LotArea	I	4	Total Area of Lots
BldgArea	I	4	Total Building Area
ComArea	I	4	Total Commercial Space Area
ResArea	I	4	Total Residential Space Area
OfficeArea	I	4	Total Office Space Area
RetailArea	I	4	Total Retail Space Area
GarageArea	I	4	Total Garage Area
StrgeArea	I	4	Total Storage Space Area
FactryArea	I	4	Total Factory Space Area
OtherArea	I	4	Total Other Area
NumBldgs	S	2	Number of Buildings
UnitsRes	S	2	Number of Residential Units
UnitsTotal	S	2	Total Number of Units
Lots	S	2	Total Number of Lots
Source	C	30	Source of Flows
SearchB1	R	8	Bus Search Radius 1
SearchS1	R	8	Subway Search Radius 1
SearchB2	R	8	Bus Search Radius 2
SearchS2	R	8	Subway Search Radius 2
Exclude	C	1	Exclude Zone From Model
Alpha	F	4	Allocation coefficient
Beta	F	4	Allocation coefficient
GammaS	F	4	Allocation coefficient - Subway
GammaB	F	4	Allocation coefficient - Bus
CentroidNode	I	4	Centroid Node ID
Dist	R	8	Distance to

We extracted building footprints from the NYCMap data and created a citywide database. A small portion of these include building names.

We also created GIS layers of Hydrography and Open Space from NYCMap and NYC Landmarks from OTIS.

Finally, we have compiled a set of Digital Orthophoto Quadrangles (DOQ) that cover the entire area serviced by the MetroCard system. These serve as a spatial reference to create accurate geographic locations for all the GIS layers created. The DOQs include the 2000 files from NYCMap for New York City, the 1999 and 2001 files from NYS Digital Orthoimagery Program for Westchester and Nassau Counties and the 2002 files from New Jersey Geographic Information Network for parts of New Jersey.

CTPP Part 3 Data

The Census Transportation Planning Package is a special tabulation of 2000 Census data to provide transportation planners detailed information from the long form. Part 3 consists of Journey-to-Work flow tables. We processed the data to produce a TAZ to TAZ matrix for the New York Metropolitan region with a separate matrix core for each principal mode of transportation to work. The flows are from Census Block Group of residence to Census Block Group of work. The transit trips matrix was thought to be a useful point of comparison for the trip tables generated from the project although it was thought that Census data would approximate a lower bound rather than a good estimate of these flows.

Senior MetroCards

Caliper requested and received the MetroCard serial numbers and mailing addresses for the users of the senior reduced fare program, agreeing to keep them confidential. The Microsoft Access database contained almost 500,000 records. Some of seniors live outside of the city, since one does not have to be a resident to participate in the program. All the records were geocoded using LION and TIGER streets (see Table 27). The addresses proved useful in helping to locate buses, since it is strongly presumed that riders utilize the closest stops to their homes, assuming they live in the city. Given the sensitive nature of this information, the address information is being tightly controlled.

Table 27: Senior MetroCard Address Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Identifier
AFC	I	4	MetroCard Fare Card Serial Number
ADDR1	C	35	Maling Address, Line 1
ADDR2	C	50	Mailing Address, Line 2
CITY	C	30	City Name
New City	C	30	Corrected Borough Name
STATE	C	2	State Abbreviation
New State	C	2	Corrected State Abbreviation
ZIP	C	5	ZIP Code
New ZIP	C	5	Corrected ZIP Code
Method	C	10	Method used to locate address
			2003 – 2003 TIGER/Line with ADDR2
			2003-A1 – 2003 TIGER/Line with ADDR1
			2003-CS – 2003 TIGER/Line with CITY/State
			2003-New – 2003 TIGER/Line with Corrected City/State
			2003-NewQ – 2003 TIGER/Line with Corrected ZIP
			LION – LION Streets
			None – Could not be located, omitted
			Outside – Outside of NYC, omitted
			POB – PO Box, omitted
Longitude	I	4	Geographic Coordinate
Latitude	I	4	Geographic Coordinate

Data Processing

This section describes procedures used to process the MetroCard and Bus Trip data, transforming each day's worth of transactional data into two tables for use with the query software: linked passenger trips and component legs for the trips.

The initial conversion and processing of the input datasets was described in the previous section. The overall flow of data is depicted in the flowcharts contained in Figure 3 and Figure 4. All the steps are documented in the following sections, concluding with documentation for using the administrative toolbox that allows an expert user to process additional day's worth of data.

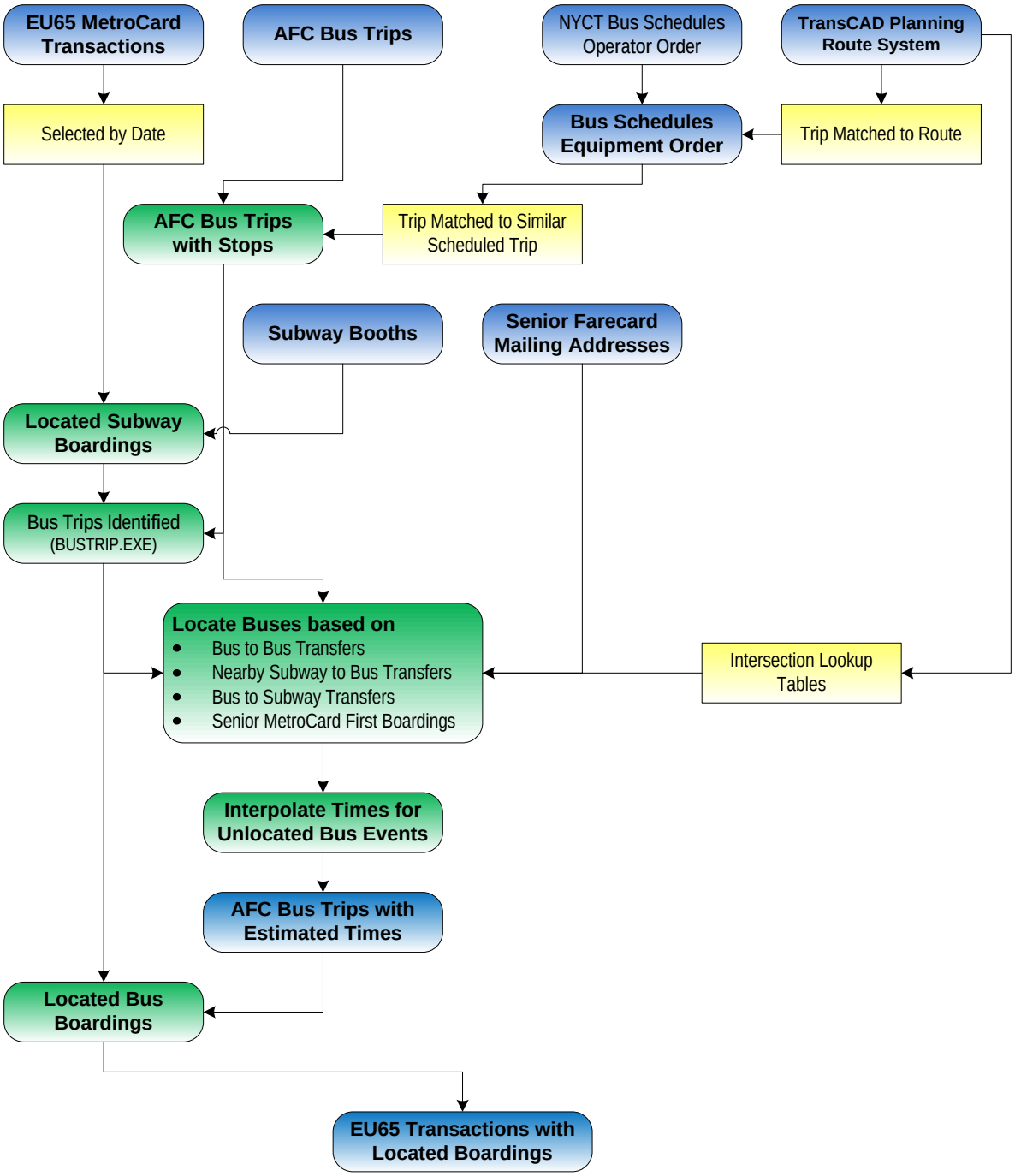

Figure 3: Data Flow (Part 1)

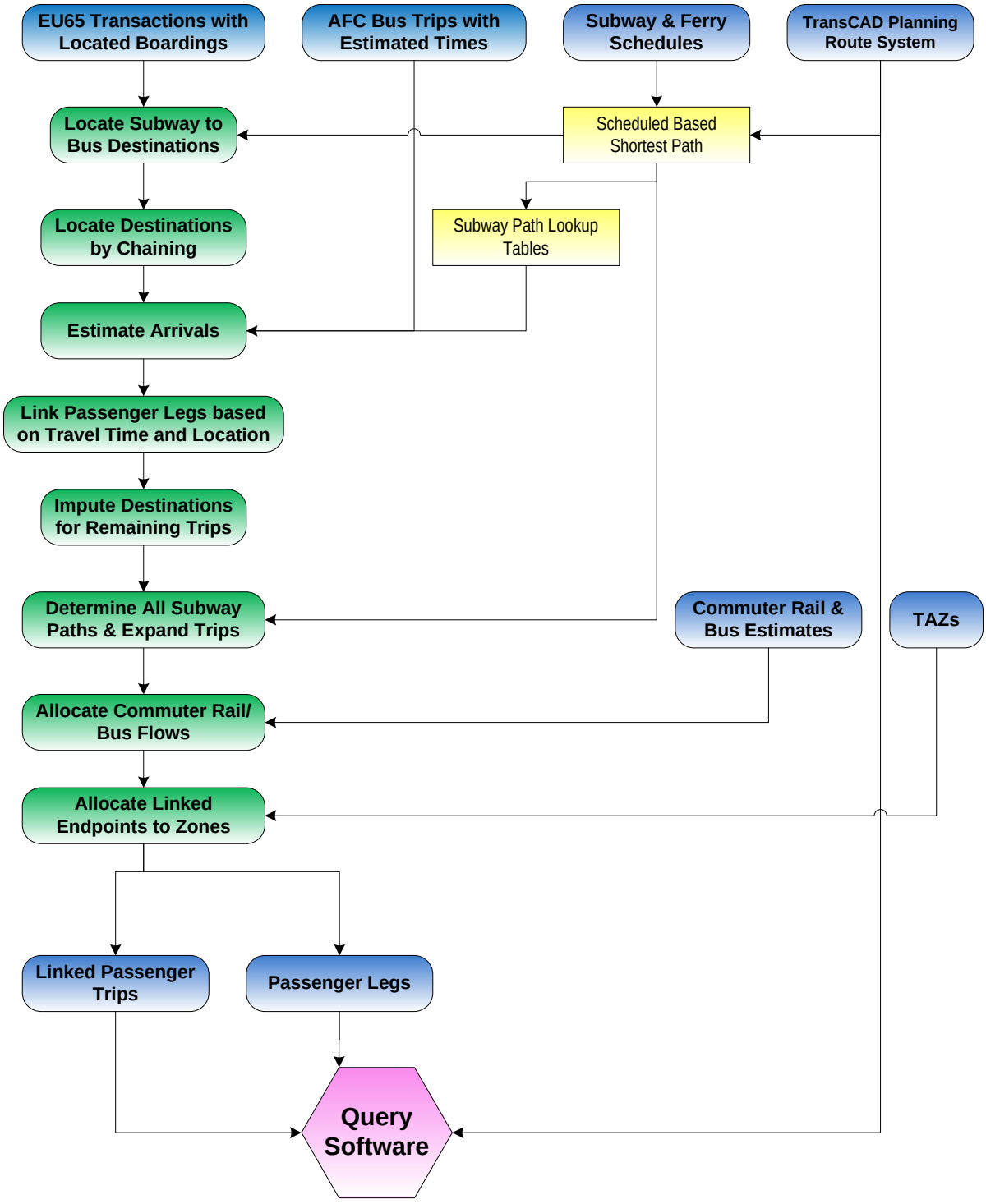

Figure 4: Data Flow (Part 2)

Lookup Table Creation

In this section, we document six lookup tables that our script creates for use in the processing of MetroCard transactions.

1. A table is created of the attributes from the stops layer of the route system for later queries. This eliminates the need to have the full route system loaded for many operations that only need access to the route stops.

Table 28: Inputs\RS_STOPS.BIN Data Dictionary

Field	Type	Length	Description
Stop_ID	I	4	Unique Stop Identifier
Longitude	I	4	Geographic Coordinate
Latitude	I	4	Geographic Coordinate
Box	I	4	DOT Box Number/Station ID
PStop_ID	I	4	Physical Stop Identifier
Route_ID	I	4	Route Identifier
Route_Name	C	32	Route Pattern Name
Station/Location	C	50	Station Name/Street Intersection
Milepost	R	4	Milepost along Route
Dir	C	1	Route Direction
Used	C	1	Used by scheduled trips
Sch_Seq	F	4	Sequence within Schedule Pattern

2. Using the information in the bus route pattern table (see Table 25), we then create a lookup table with the key *Carrier Code:Location Code:Destination Sign*. The table provides the matching TransCAD route ID and the number of distinct paths observed. This is used to help determine the route matching a bus trip.

Table 29 Inputs\Bus_Route.Bin Data Dictionary

Field	Type	Length	Description
Key	C	13	Search Key *Carrier Code:Location Code:Destination Sign*
Route_ID	I	4	Route Identifier
NPaths	I	10	Number of distinct paths observed

3. Using the TransCAD route system, we create an intersection lookup table from any route to a bus route. The key is *Bus Route ID:Prev Route ID*. The table provides the closest bus stop ID within .3 miles, along with the box number and geographic coordinate. The alighting stop id is also provided. This table is used to identify transfer points that can be used to pin down the location of buses.

Table 30: Inputs\Stops.Bin Data Dictionary

Field	Type	Length	Description
PRoute_ID	I	4	Previous Route Identifier
Route_ID	I	4	Route Identifier
Stop_ID	I	4	Stop Identifier of Closest Bus Stop
PrevStop_ID	I	4	Previous (Alighting) Stop Identifier
Bus_Key	C	9	Search Key *Bus Route ID:Prev Route ID*
Longitude	I	4	Geographic Coordinate of Closest Bus Stop
Latitude	I	4	Geographic Coordinate of Closest Bus Stop
Box	I	4	DOT Box Number of Closest Bus Stop
PStop_ID	I	4	Physical Stop Identifier of Closest Bus Stop
Subway	C	10	"S" indicate previous subway route
N_Int	S	2	Number of intersections between the two routes

4. Using the TransCAD route system and the All to Bus table from step 3, we create an intersection lookup table from a boarding subway station to bus route. The key is *from_physical stop:bus route* and the table provides the boarding stop, physical stop ID and box number and the distance traveled along the subway system. Only trips up to five miles with a unique intersecting bus stop are included. This table is used to identify subway to bus transfer points.

Table 31: Inputs\P2B5.Bin Data Dictionary

Field	Type	Length	Description
FromPStop	I	4	Previous Physical Stop Identifier (Subway Station)
ToRoute	I	4	Bus Route Identifier
ToStop	I	4	Stop Identifier of Closest Bus Stop
ToBox	I	4	DOT Box Number of Closest Bus Stop
ToPStop	I	4	Physical Stop Identifier of Closest Bus Stop
K	C	10	FromPStop + ":" + ToRoute
Miles	R	8	Distance (miles) between Station and Bus Stop
Longitude	I	4	Geographic Coordinate of Closest Bus Stop
Latitude	I	4	Geographic Coordinate of Closest Bus Stop

5. Using the TransCAD route system, we create an intersection lookup table from any bus route to a subway station. The table provides the closest bus stop ID within .3 miles. The key is *route|00station* and the table provides the physical stop ID, box number and coordinate of the alighting bus stop. This table is used to identify bus to subway transfer points.

Table 32: Inputs\B2S.Bin Data Dictionary

Field	Type	Length	Description
Route	I	4	Bus Route Identifier
Station	I	4	Subway Station Identifier
Stop	I	4	Stop Identifier of Closest Bus Stop
Box	I	4	DOT Box Number of Closest Bus Stop
PStop	I	4	Physical Stop Identifier of Closest Bus Stop
Longitude	I	4	Geographic Coordinate of Closest Bus Stop
Latitude	I	4	Geographic Coordinate of Closest Bus Stop
Key	C	11	Route + "|00" + Station

6. Lastly we create a lookup by origin booth and destination subway station node by using TransCAD's schedule-based shortest path (SSP) procedure that Caliper has previously developed for transit customer information system applications. This allows us to estimate travel times between each pair of stations and to correct the likely alighting station when there are multiple stations nearby servicing different subway lines, e.g., Whitehall and Bowling Green. The table allows us to quickly determine a reasonable path through the subway system, including the estimated travel time.

   We start by identifying alternate stations within a mile of each station, keeping up to five; this is stored in the stations.cdf layer (see Table 20). The NearN*n* and NearS*n* fields contain the station northbound and southbound physical stops. Time*n* contains the walking time in hours from the queried station to the nearby station *n*.

   We create a lookup between pairs of stations with a SSP query for each hour of the day. The origin is specified by Booth code and the destination by station node ID. Table 33 documents the query table. Table 34 documents the path detail table.

   The SSP procedure requires a route system with all the possible patterns, a complete list of stations and a schedule with all trips. To that end, we processed the 2004 RTIF subway schedules and converted them into TransCAD format. We then matched the patterns against the existing routes in the TransCAD route system and added all the missing subway patterns and variations; we also added the variations for the SI Railway. A subset of the route system consisting of the subway, RI Tramway, SI Ferry and SI Railway and PATH routes was created and combined with the subway, tramway, ferry, SIR and PATH schedules and loaded into the SSP.

   The SSP queries allow us to implicitly handle the SI Ferry, since queries starting outside of Staten Island and ending there must use the ferry for their itineraries.

   The SSP uses 3mph as the standard walking speed. We had to decrease its affinity for walking by adding a fixed penalty, because the SSP was skipping short final transit legs that are used in practice, e.g., Park Place on the Franklin Avenue Shuttle.

Manual edits can be made to the lookup table if it is determined at a later date that a better path exists or an adjustment is required for some other reason.

Table 33: Subway SSP Query Table

Field	Type	Length	Description
ID	I	4	Unique query identifier
Orig	I	4	Origin unique identifier
Orig Booth	C	5	Origin Booth Code
Orig NB_PStop	I	4	Origin Northbound Physical Stop Identifier
Orig SB_PStop	I	4	Origin Southbound Physical Stop Identifier
Dest	I	4	Destination unique identifier
Dest Node	I	4	Destination Station ID
Dest NB_PStop	I	4	Destination Northbound Physical Stop Identifier
Dest SB_PStop	I	4	Destination Southbound Physical Stop Identifier
Minutes	I	4	Query Departure Time (Min past midnight)
#Subway Legs	I	4	Number of legs in subway path
Final Walk Time	R	4	Walking time (min) to queried destination station
Orig Stop	I	4	Origin Route Stop Identifier
Dest Stop	I	4	Destination Route Stop Identifier
SIF Stop	I	4	Last Subway Route Stop Prior to SIF
Orig Time	R	4	Departure Time (min) of first subway train
Dest Time	R	4	Arrival Time (min) of last subway train/ferry
Orig Route	I	4	Origin Route Identifier
Dest Route	I	4	Destination Route Identifier
Dest Longitude	I	4	Destination Geographic Coordinate
Dest Latitude	I	4	Destination Geographic Coordinate
Dest Box	I	4	Destination Station Identifier
Dest PStop	I	4	Destination Physical Stop Identifier
Dest Dir	C	1	Destination Route Direction (N/S)
Key	C	15	[Orig Booth] + "|" + [Dest Node] + ":" + Minutes
Subway_Path	I	4	Subway Path Identifier into path detail table

Table 34: Path Detail Data Dictionary

Field	Type	Length	Description
Path_ID	I	4	Subway Path Identifier
Sequence	I	4	Leg Sequence Number
Orig Stop	I	4	Leg Origin Route Stop Identifier
Orig Time	I	4	Leg Boarding Time (min)
Dest Stop	I	4	Leg Destination Route Stop Identifier
Dest Time	I	4	Leg Alighting Time (min)

AFC Bus Trip Processing

The AFC bus trip table logs information from actual trips, but contains no location or stop information. After the conversion and cleanup phase previously described, there should be

one record present for each actual trip; this is not always true due to the quality of the input data. In this section, we describe the steps that indicate how an extract of the bus trips is created for the day being processed. The extract is then matched against the schedule to find a similar scheduled trip for each actual trip. The schedule is used to provide the sequence of stops and approximate times.

1. Select a subset of trips, keeping only revenue trips with non-zero durations that we have identified and that span any of the 3am-3am day being processed. For NYCT, this is sign codes 1000-8999. For DOT, this excludes sign codes 4001-4021.

2. For NYCT buses, identify the STIF path as follows:
    a. For sign codes with a unique path, assign that path.
    b. For trips other than the initial one, consider the previous sign code paired with the sign code and assign paths when there is a unique correspondence with the pair of sign codes. This rule was derived from the idea of using the previous trip's sign code to determine the correct path.
    c. If there are multiple paths that match the previous sign code/sign code pair in step (b) then choose the one with the most scheduled trips that:
        i. Has a duration within 10% of the AFC trip duration.
        ii. If none exist, then has a duration exceeding the AFC trip duration.
        iii. If none exist, then the path with the most scheduled trips.
    d. For the remaining records, use a key based on the depot code, run number without the route and the sign code to match with the STIF path servicing that combination of codes. Use the logic from step (c) to break ties based on the duration and the most scheduled trips.
    e. For the remaining records, use a key based on the depot code and the sign code to match with the STIF path servicing that combination of codes. Use the logic from step (c) to break ties based on the duration and the most scheduled trips.
    f. For the remaining records, use the sign code to match the STIF path servicing that sign code. Use the logic from step (c) to break ties based on the duration and the most scheduled trips.

3. For DOT/LIB buses, identify the path using step (a).

4. Determine a reasonable trip based on the pattern code (PATH), run number, sign code and time, by minimizing the sum of the absolute differences between the actual and scheduled starting and ending times. NYCT, LIB and DOT trips are processed separately.

5. For trips longer than 180% of the scheduled trip or at least five minutes longer, shorten the trip record to 110% of the scheduled trip. This handles the infrequent situation where the driver failed to change the sign code and the trip record exceeds the expected duration.

Processing of EU65 Transactions

This section describes how we process the EU65 MetroCard transactions to create located, linked passenger trips. We begin with an overview of the location methodology and then provide a more detailed description of the procedure.

Locating Subway Transactions

For location purposes, we consider any EU65 transaction for a transit mode with fixed booth locations to be a subway transaction. These include the NYCT subway, the Staten Island Railway, the Roosevelt Island Tramway and the Port Authority's JFK AirTrain and PATH train.

For each subway transaction, we use its booth code to determine the appropriate station. The transaction is located to the geographic coordinate associated with the station in the TransCAD route system based on NYCMap.

The vast majority of transactions occur prior to entry and provide a boarding location and time for the passenger. For the Staten Island Railway northbound and passengers leaving JFK Airport on the AirTrain, the transaction provides an alighting location and time.

Locating Bus Transactions

The largest unanticipated expenditure of time in performing this project was trying to figure out a method to clean up the bus trip tables and match them against the schedule, so that they could be used to locate the bus boardings. The principal difficulty was the inconsistent number of records corresponding to an actual trip. In order to create a single trip record, there is a frequent need to combine multiple trip records. In other cases, records need to be split to correct for a driver's failure to change the sign code. We had several false starts before we settled on our final strategy.

Our original strategy to locate bus transactions involved cleaning up the AFC bus trip table to contain one record per actual trip. The scheduled trip, along with the TransCAD route identified using the schedule files, would control the cleanup process. The scheduled stop times would be adjusted to reflect the actual times observed and passenger dips would be located based upon interpolation along the trip route.

Unfortunately, a variety of data imperfections and matching issues combined to foil our cleanup and matching strategy. After several false starts and some valuable suggestions made by Larry Hirsch, we developed a successful approach based on the following observations:

1. The most important goal of matching the bus trip to the schedule is to determine the route pattern, and the location of and relative times between stops. Thus, it is sufficient to find a similar trip made by a bus, as long as it uses the same pattern. It is

no longer problematic if two trips match the identical scheduled trip, since only the pattern of stops and their relative times are used by the location procedure.

2. Bus positions can be localized using transfers observed in the data. For example, if a passenger takes two bus legs within a short period of time and the routes intersect then the second bus must be near the intersection point at the transaction time. Similarly, some subway to bus locations can be used as well.
3. Home addresses of participants in the discount Senior MetroCard program allow the identification of the nearest bus stop(s) to their residences. In many cases, these stops can help pin down bus locations.

We now describe the location procedure for both subway and bus transactions. Each major step starts with an overview discussion and then details the steps involved. This documents the procedure developed to locate or impute alighting locations and create linked passenger trips.

We start by employing a strategy we call *chaining.* Each transaction defines a *leg* (unlinked trip) in the passenger's journey. The location of the next transaction by a MetroCard is used to find a nearby bus stop or subway station for the alighting location of the current leg, assuming a consistent one exists. An example where an inconsistency arises is when a subway leg is followed by a bus dip and no subway station is near the bus stop, so we cannot determine the alighting station. We loop back to the first transaction from the final leg, unless the passenger only makes a single linked trip during the day. This logic is justified by the fact that many riders return to their origin at the end of the day,

We estimate the arrival time for each assigned alighting location using either a lookup table or running a SSP query for each subway journey or using the estimated stop times computed for the bus trip. We combine legs into a single linked trip if the expected arrival time of the first leg is within eighteen minutes of the start of the next leg. The location consistency is ensured by the fact that we have not yet assigned alighting locations for inconsistent legs.

Not all transactions will have their alighting locations determined by the chaining procedure, since it fails to assign an alighting location when either the rider made a single trip (which could possibly be a multi-modal trip) or no alighting stop is consistent with the next boarding location. We then use two expansion procedures that use sampling to assign alighting locations. For subway transactions, we assign an alighting stop by uniformly sampling based on the observed distribution from riders boarding at the same station with assigned alighting stations. For bus passengers, we use a similar approach based on a distribution of alighting stops for all passengers boarding at the same stop during the day for that route pattern.

Multiple riders on a cash-based MetroCard were collapsed into a single record previously so their transactions should not create any erroneous trips.

We use a two pass method to locate bus transactions. During the first pass (steps 15-17), our primary goal is to pinpoint the location of the bus at several stops during each trip by using bus to bus and subway to bus transfer points and senior home stops. Interpolated times are

then assigned to intermediate stops and these are used to assign locations to the remaining bus dips during the second phase (step 20). Adjustments to the boarding stop used are made during our trip linking procedure.

We originally suggested that a bus stop group be used for bus locations, due to the anticipated inaccuracies with the data. Unfortunately, no previous definition exits, nor is there any obvious one. Instead, we assigned stops by sampling based on a uniform distribution among the stops with the same estimated time during a bus trip. The distribution used can be adjusted by changing the weight field in the table of STIF pattern stops, from the default value of one.

There are two aspects of the EU65 data collection process that decrease the locational accuracy that can be achieved. First, the EU65 transaction times are truncated to six minute accuracy. Second, the boarding order is not maintained between the collection points and the extract that we received from the mainframe computer. The second issue precludes reducing the locational uncertainty for boardings with the same time as transfer points.

Detailed Description of MetroCard Transaction Processing

This section describes in detail, how the MetroCard transactions are processed to create the two tables (linked passenger trips and component trip legs) for the query software. The twenty two steps described below correspond to macros in the processing script.

1. **Select the transactions for a day and perform some preliminary processing.**
    a. Start with the two week EU65 file previously described. For subway/tram transactions, the booth code has already been used to locate the transaction and we mark such transactions with From_Method=”F”. For SIR northbound, we will change this to an alighting location in step f. Select the EU65 transactions for the specified day and create a subset table with some extra fields, including minutes past midnight for the transaction time.
    b. Consolidate records corresponding to multiple passengers using a value-based MetroCard into a single record and populate the Riders field. Otherwise, Riders is set to one.
    c. Combine records for the Howard Beach JFK AirTrain/Subway Train $7 combined fares, changing the booth code to JFKS2 to identify the outbound direction of travel.
    d. Add carrier codes for bus transactions and change the mode code to “E” for Express Bus Trips using the location code
    e. Add destination information for tramway transactions. Such transactions are marked To_Method=”F”.
    f. Correct SIR Northbound transactions to have their destination located, but not the origin. Mark them with To_Method = “F” and clear From_Method.
    g. Use bustrip.exe procedure, developed for Task 2, to add bus trip ids and sign codes to the transactions. This uses the bus number and location code and time interval to identify the bus trip and is successful 97% of the time.

h. For most of the remaining unmatched records, a second pass by bustrip.exe adds the bus trip ids and sign codes by using just the bus number and time interval and ignores the location code. This seems to be reasonable, since there can be errors in the sign code to location code correspondence tables in the bus fare boxes. A small number of transactions cannot be matched to any trip.
i. Add service message sign codes based on the carrier and location codes (e.g., Next Bus Please)
j. For bus transactions, add the boarding route ID using a carrier/loc/sign based key into a lookup table of bus route patterns
k. For subway/tram transactions, add the boarding northbound and/or southbound physical stop IDs using a lookup table accessed via the booth code.
l. Add transfer point boarding locations for linked bus to bus trips for the subsequent leg. The route stop, physical stop id and box number are filled in. We consider the transactions to be linked if they occur within thirty minutes of each other. Sixty minutes is used if the first trip was on an express bus. Mark such transaction From_Method = "I" and Linked="3" (30 minutes) or "6" (60 minutes).
m. For simple linked subway to bus transactions, add the transfer point boarding location as above. This is repeated twice: once for northbound and once for southbound stops. Our restrictive conditions limit this to when all the subway paths out of the stop intersect with the linked bus route at the same location or not at all. Additionally, the subway trip length cannot exceed five miles. The trips need to be short since we do not want any subway-to-subway transfers to occur. Again a thirty minute restriction is used. Mark such transactions From_Method = "N" or "S".
n. Add bus boarding locations for the first dip by a senior Metrocard within .5 miles of their home address by using the closest bus stop on the identified route. These are the *senior home* stops. The senior MetroCard must have been used at least twice that day, since it is less certain that a single transaction occurred near the senior's home. Mark such transactions From_Method = "O".
o. Add bus alighting locations for stops based on linked transfers to the subway within 30 minutes when the closest stop is within .3 miles of the subway station. Mark such transactions To_Method="I".

2. Estimate times for individual bus trips and locate bus boardings.

This step uses the methodology described above to locate buses by using transfer points and senior home stops. Bus boardings are then located using the estimated locations.

a. Sort the transactions by bus trip id and transaction time.
b. For each bus trip:
    i. Identify the corresponding passenger transactions.
    ii. Load the list of schedule stops with scheduled times for the associated trip.
    iii. Associate the start and end bus trip time with the first and last stops, converting the times to tenths of an hour.
    iv. Associate the transfer point/senior home stops with their transaction times using the DOT box number.

v. Discard intermediate times from their stops when they are inconsistent with the neighboring stops with times.
vi. Interpolate between stops based on the scheduled times, converting to tenths of an hour. At this point, each stop has an estimated time.
vii. For each transaction time value, determine the list of possible stops.
viii. For each passenger transaction, randomly select a stop from the possible list of stops using a uniform distribution. This labels the transaction with a route ID and DOT box number, which will later be expanded to include the route stop id, physical stop id and route milepost. If the transaction time does not match a stop (e.g., falls in a twelve minute gap between stops), then use the last stop prior to the gap. If the transaction time is earlier than any stop then use the first stop. Mark such transactions From_Method = "i".

3. Perform SSP queries for subway to bus transitions.

This step creates a table of additional SSP queries with the actual transaction times when we observe a subway leg followed by a bus leg. This provides more accurate estimates for the subway trip. The setup is the same as used to create the lookup table in step 6. The five nearest subway stations within a mile of the bus boarding location are considered. The answer returned must arrive no later than ten minutes after the bus transaction for it to be used.

4. Update the subway to bus linkages based on the SSP queries.

This step updates the subway to bus leg linkage based on the SSP queries in the prior step.

a. If a subway arrival is within eighteen minutes of the bus transaction, consider the trips to be linked and mark Linked="S". If there is no feasible path, then we know that the subway destination cannot be near the bus transaction.
b. Use the shortest path answer to assign boarding and alighting subway routes and stops. The alighting time is also estimated from the results and includes the walking time. This logic is used even when the trips are not linked due to excessive time. Mark such transactions To_Method="S".
c. If the distance from the subway alighting stop to the bus stop is at most .5 miles, then snap the bus stop to the stop nearest the alighting station if necessary. Add the From_Method suffix of "S" to stops which have been snapped.

5. Fill in coordinate information for bus transactions with partial stop information.

For bus transactions with partial stop information, such as box number, add coordinates when known for that DOT box.

6. Use the chaining rule to derive alighting locations.

This step performs the chaining rules that derive the alighting locations based on the subsequent boarding locations.

a. For MetroCards with more than one trip during the selected time period (day), apply the chaining rule that the destination of a trip is the stop closest to the origin

of the next trip. The first and last trips complete the chain, except when all the trips for the day have been linked into a single passenger trip. Mark chained transactions To_Method="C".

b. For subway trips not previously handled by the Subway-to-Bus case, use the SSP lookup table with the boarding booth and the station closest to the next transaction. This is the next station if it is a subway trip or the closest station within a mile otherwise. The lookup table allows us to assign the routes and stops and estimate the arrival time. Add an "L" suffix to To_Method for transactions so located.
c. For bus trip alighting stops, we assign the route stop, including milepost and physical stop, closest to the next transaction. If the route stop is not associated with the scheduled stop then use the id from the preceding or subsequent stop, if available. We require the stop to occur after the boarding stop by using the mileposts from the route system. Additionally, the stop must be within a mile of the next transaction's location.

7. Update linkage information for alightings derived by the chaining rule.

a. Estimate bus arrival times using the matched schedule trip. The time between stops is added to the boarding time.
b. For transactions with estimated arrival times, if that time is within eighteen minutes of the next transaction, then mark them as linked. Mark such transactions Linked="N".

8. Update bus to subway transitions.

For bus transactions linked to subway transactions, snap the bus alighting stop to the closest one to the station.

9. Add route milepost information.

Add starting and ending mileposts for each transaction from the stops layer, so that they can easily be mapped.

10. Assign subway alighting locations by random sampling.

For subway legs without alighting locations, choose one uniformly from the destinations assigned to other passengers using the same booth during the same time period: 3-6am, 6-10am, 10am-3pm, 3-7pm, 7-11pm, 11pm-3am.

11. Add additional fields with origin/destination information.

Assign origin and destination nodes, O and D to each passenger leg. For subway, this is the station complex node. For bus, this is the nearest street intersection. Add Origin/Destination Borough and Zone information. This will be updated for the endpoints of linked trips using the zone allocation procedure.

12. Remove degenerate trips.

Remove bus destination information where the alighting stop appears to be prior to the boarding stop

13. Assign bus alighting locations by random sampling.
For bus legs without alighting locations, choose one uniformly from the destinations assigned to other passengers using the same route stop during the same time period. The route stop implies that their trip followed the same pattern.

14. Perform SSP queries for all subway journeys.
Refine the subway paths by performing a new SSP query for each subway trip using the actual departure time a random time between 0 and 5 minutes, uniformly chosen, to get more accurate subway paths. The queries are sorted for more efficient processing.

15. Expand subway records into multiple legs based on the SSP queries.
Use the results from the prior step to expand the MetroCard transactions in a passenger leg table. Each subway journey becomes one or more legs depending on the SSP path.

16. Compute the sampling rates for commuter flows to subway.
Certain subway stations and bus stops have significant flows of commuters arriving from MetroNorth, LIRR and NJT. These flows need to be specially handled so that they are not assigned to the immediately adjacent zones, distorting the true origins for the trips. We received estimates of the significant commuter flows from NYCT and collected them in the Excel spreadsheet CRflows.xls. In each case, we estimated the forward and reverse flows by time period and mode (subway, bus).

This step uses actual MetroCard transactions by time period (6-10am, 10am-3pm, 3-7pm, night) to allocate the flows between multiple subway stations when more than one services a commuter rail/bus station.

We then compute sampling rates for each station for boarding and alightings to assign the correct proportion of traffic to the artificial commuter zone. We exclude the joint-CR and PATH MetroCards from the sampling rate computations. The boarding rate for a particular station during a time period is computed by starting with the number of estimated commuters, subtracting the observed trips on that day for joint-CR MetroCards and dividing the result by the number of non-joint-CR trips. The alighting rate is computed in a similar fashion.

17. Allocate commuter flows to nearby bus stops based on the number of trips servicing each stop.
Bus stops are allocated for commuter flows based on the number of trips servicing the nearby stations during a time period, instead of the total boardings/alightings. This step computes the number of trips.

18. Compute the sampling rates for commuter flows to bus.
This step computes the sampling rates for buses, in a similar fashion to step 16, except it uses the bus allocation by trip computed in step 17.

19. Create lookup tables of stations/bus stops that service commuter rail/bus.

20. Assign explicit commuter flows to external zones.
Use the lookup tables from step 19 to assign transactions from joint-CR tickets to the appropriate artificial zone. Joint-CR tickets are combination tickets for MetroNorth or LIRR and NYCT. We also look at MetroCards that were used at the PATH WTC station. We consider these to be joint-PATH tickets and assign them to external PATH zones when they are used near a PATH station.

21. Allocate endpoints of linked trips to origin/destination zones.
This step assigns the endpoints of linked trips to a nearby zone, unless they have already been marked as a commuter rail or bus transaction.

a. Determine the list of adjacent zones for each station and bus stop (box).
b. For each zone, select all the stops inside or within distance d_1 of the zone. Select additional stops within distance d_2 of the zone if their routes has no other stops previously selected.
c. d_1 and d_2 were set by borough, but can be changed by zone. Separate values are used for local bus and subway/express bus (miles):

Table 35: Search Distances by Borough and Mode

Borough	Local Bus d_1	Subway d_1	Local Bus d_2	Subway d_2
Lower Manhattan	0.05	0.20	0.15	0.35
The Bronx	0.05	0.30	0.20	0.50
Brooklyn	0.05	0.25	0.20	0.50
Queens	0.05	0.30	0.25	0.50
Staten Island	0.05	0.30	0.25	0.50
Upper Manhattan	0.05	0.25	0.20	0.50
Nassau	0.05	0.30	0.35	1.00
Suffolk	0.05	0.30	0.50	1.00
Westchester	0.05	0.30	0.25	0.50

d. For each station/bus stop, collect the list of zones that include the station/stop. These will be considered for zone allocation.
e. Excluded, no-trip zones were omitted from the first step, so they cannot be adjacent to a station/stop.
f. Compute a skim matrix with the distance from each zone centroid to the intersection nodes adjacent to all the subway stations and bus stops, using the street network to compute walking distance.
g. Process the linked trip table to allocate each origin and destination to a zone. We will describe the method for subway origins first and then describe the changes necessary for destinations and other modes.
h. Trips originating at subway stations are allocated to zones by considering the adjacent zones previously determined. The probability that an origin zone is selected is based on a combination of the zone's population, employment and the walking distance between its centroid and the station. A logit model is used for the distance portion to

penalize longer walks. We use a combination of the zone's Population aged 15+ from the 2000 Census SF1 and the CTPP employment, that varies based on the time period. For 6-10am and at night, we use just the population. During midday (10am-3pm) we average the population and employment. During 3-7pm, we use the employment.

i. For station i and zone j, we determine its weight w_j as $[\text{Normalized Population}]_j * e^{\gamma * d_{ij}}$ or 0 if the zone has no population. Populations are normalized so that the sum of all zones near station i equals one. Gamma, γ was determined by borough, but can be set by zone:

Table 36: Weighting Parameters

Borough	Local Bus γ	Subway γ
Lower Manhattan	-2.03	-3.10
The Bronx	-1.31	-2.34
Brooklyn	-1.31	-2.34
Queens	-1.31	-2.34
Staten Island	-1.00	-2.03
Upper Manhattan	-1.72	-2.77
Nassau	-1.31	-2.34
Suffolk	-1.31	-2.34
Westchester	-1.31	-2.34

j. The probability that zone j is chosen, p_j, is its weight w_j divided by the sum of the weights of all zones near station i.
k. Destinations are allocated in a similar fashion, substituting CTPP employment for population. However, at night we still use the population.
l. Express buses are handled with the same gammas as the subway, using the node closest to the DOT bus stop for the distances.
m. Local buses have their own gammas, since riders are less likely to walk far to get to a local bus compared to a subway station.

22. Reformat the tables for the query software.

This step performs the final formatting to create the linked trips and component legs tables for the Query software. This includes adding/removing some fields and indexing. The tables are documented in Tables 37 and 38.

Table 37: Passenger Leg Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Record ID
Serial	I	4	MetroCard Fare Card Serial Number
Type	C	3	Transaction Type
Class	C	3	MetroCard Class (Fare Media)
Mode	C	1	Transit Mode: B, E, F, I, J, P, S, T
Booth	C	5	Booth Number/Authority Code
Unit ID	C	4	Station Controller Computer/Depot
Point of Entry	S	2	Point of Entry
Carrier	S	2	Numeric Carrier Code
Authority	C	2	Authority Code
Bus#	I	4	Bus Number (From SCP)
Sign	I	4	Destination Sign Code
Trip	I	4	AFC Bus Trip ID
Riders	S	2	Number of Riders
Departure	I	4	Departure Time (Minutes past midnight)
Route	I	4	TransCAD Route ID
From_Stop	I	4	Boarding TransCAD Stop ID
From_Box	I	4	Boarding NYCDOT Box
From_Pstop	I	4	Boarding TransCAD Physical Stop ID
From_Method	C	2	Method used to locate boarding
From_Milepost	F	4	Boarding Milepost along TransCAD Route
O	I	4	Boarding Node (Station Complex or Nearest Street Intersection)
OTAZ	C	11	Boarding Zone
OTract	C	11	Boarding Census Tract
OCounty	I	4	Boarding County(Borough) FIPS Code
Arrival	I	4	Estimated Arrival Time (Minutes past midnight)
To_Stop	I	4	Alighting TransCAD Stop ID
To_Box	I	4	Alighting NYCDOT Box
To_Pstop	I	4	Alighting TransCAD Physical Stop ID
To_Method	C	2	Method used to locate alighting
To_Milepost	F	4	Alighting Milepost along TransCAD Route
D	I	4	Alighting Node (Station Complex or Nearest Street Intersection)
DTAZ	C	11	Alighting Zone
DTract	C	11	Alighting Census Tract
DCounty	I	4	Alighting County(Borough) FIPS Code
Linked	C	1	Whether the leg continues a linked passenger trip
NLegs	S	2	Number of legs in the linked passenger trip
Leg	S	2	Leg sequence number in the linked passenger trip

Table 38: Linked Trip Data Dictionary

Field	Type	Length	Description
ID	I	4	Unique Record ID
Serial	I	4	MetroCard Fare Card Serial Number
Class	C	3	MetroCard Class (Fare Media)
Mode Pattern	C	1	Transit Modes Used by Linked Trip
Riders	S	2	Number of Riders
Departure	I	4	Departure Time (Minutes past midnight)
ID1	I	4	ID of First Leg of Linked Trip
O	I	4	Origin Node (Station Complex or Nearest Street Intersection)
OTAZ	C	11	Origin Zone
OTract	C	11	Origin Census Tract
OCounty	I	4	Origin County(Borough) FIPS Code
Arrival	I	4	Estimated Arrival Time (Minutes past midnight)
D	I	4	Destination Node (Station Complex or Nearest Street Intersection)
DTAZ	C	11	Destination Zone
DTract	C	11	Destination Census Tract
DCounty	I	4	Destination County(Borough) FIPS Code
NLegs	S	2	Number of legs in the linked passenger trip

Trip Matrices

There are two standard trip matrices that are commonly used by OP for planning. We created macros to create these matrices and the methods are documented in this section. They can be created using the Create Trip Matrix checkbox in the Administrative Toolbox

AM Peak Trip Matrix

This matrix contains the AM Peak (6-10am) zone-to-zone ridership, including zones from outside of NYC. The macro creates a square matrix from the Block Group layer and aggregates the trip ridership from the linked trip table for trips with departures from 6am inclusive to 10am exclusive. The matrix is created in the WIP subdirectory with the filename MMDDYYam.mtx. This matrix can also be created with the Query software, but this macro is more efficient.

AM Peak Subway Matrix

This matrix contains the AM Peak Hour Subway station-to-station ridership. NYCT OP uses a peak hour definition that varies by subway station to attempt to capture trips entering the Manhattan CBD during the peak hour. Based on this departure hour, which is really 66 minutes, we aggregate subway trips to the station-to-station matrix. We recombine the expanded subway legs into subway journeys to avoid counting the internal subway transfers in this matrix. The matrix is created in the WIP subdirectory with the filename

MMDDYYam_subway.mtx. This matrix cannot be created with the Query software due to the special handling for complete subway journeys.

Administrative Toolbox

The Administrative Toolbox provides a simple front-end for an expert user to run most of the GISDK processing scripts created for this project (see Figure 5).

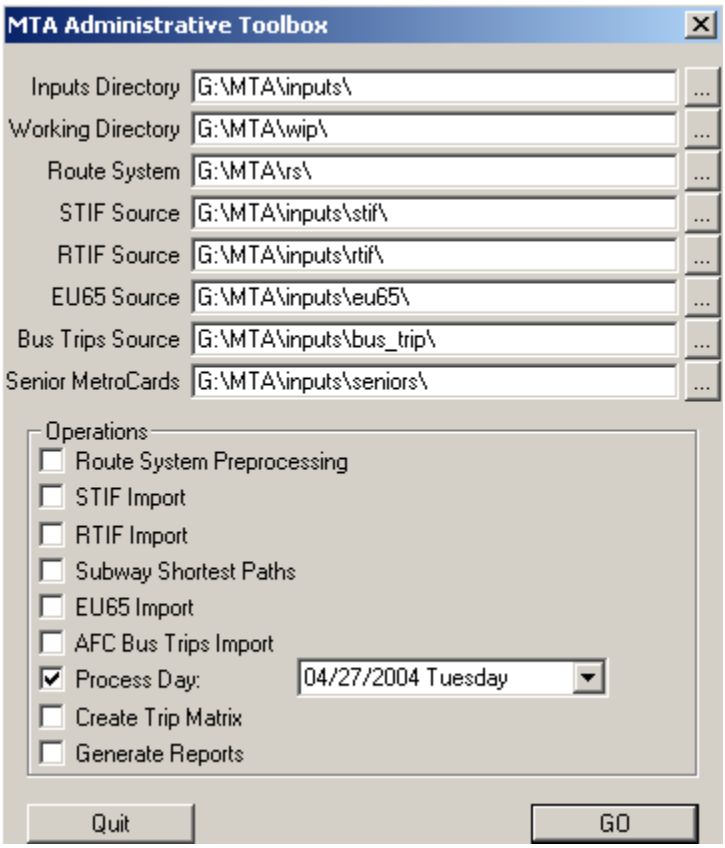

Figure 5: Administrative Toolbox

Installation

Please follow these steps to install the Administrative Toolbox:

1. Install TransCAD 5.0 from the MetroCard Query Software DVD. The User's Guide documents this process.

2. Create an MTA directory and copy the Inputs, Programs, RS and WIP subdirectories from the DVDs into this directory.

3. Add the tool to your add-ins menu by:

a. Open the Setup Add-Ins dialog box in TransCAD under the menu: Tools…Setup Add-Ins…
b. Click Add
c. Type is Dialog Box
d. Description is MTA Administrative Toolbox
e. Name is MTA Administrative Toolbox
f. UI Database is C:\MTA\PROGRAMS\ADMIN.DBD (modify to match the location of the MTA\PROGRAMS directory)
g. In Folder is None
h. Click OK

The tool can be run from Tools…Add-Ins -> MTA Administrative Toolbox. The paths to the various input datasets can be changed as needed. The desired operations are then checked off and the procedures are started once 

is clicked.

- **Inputs Directory**. The location of many of the source datasets.
- **Working Directory**. The location where temporary and output files are written.
- **Route System**. The location of the TransCAD Planning Route System.
- **STIF Source**. The location of the STIF NYCT Bus Schedules
- **RTIF Source**. The location of the RTIF NYCT Subway Schedules.
- **EU65 Source**. The location of the EU65 MetroCard Transaction files from the Mainframe.
- **Bus Trip Source**. The location of the AFC Bus Trip Tables.
- **Senior MetroCards**. The location of the Senior MetroCard mailing addresses.

Operations:

- **Route System Preprocessing**. Create lookup tables from the TransCAD Planning Route System.
- **STIF Import**. Process the STIF files, creating a table of schedule events and a table of route stops.
- **RTIF Import**. Process the RTIF files, creating a table of schedule events and a table of route stops.
- **Subway Shortest Paths**. Create a lookup table of paths between every pair of subway stations by hour using the SSP procedure of TransCAD.
- **EU65 Import**. Convert and merge all the EU65 MetroCard transactions using EU65.EXE. Some cleanup is performed on the resulting EU65 table.
- **AFC Bus Trip Import**. Convert and merge all the AFC Bus Trip logs.
- **Process Day**. This is the main processing procedure. It includes selecting the MetroCard transactions and AFC Bus Trips by the date specified and all the processing required to create the two tables for the query software.
- **Create Trip Matrix**. Create a zone-to-zone matrix of AM Peak linked trips.
- **Generate Reports**. Create a variety of reports used for validation.

Validation

A variety of methods were used to validate the location and linkage procedure while it was being developed. In particular, we implemented some automated procedures to tabulate the located transactions so that the results can be compared against other available information for validation purposes. Those procedures are now obsolete due to the powerful query software.

For subway transactions, we tabulated the entrance and exit counts by subway station complex for the full day and by four hour periods, matching the periods used at each station to match the polling of the registers; these periods vary in their starting hour. These results can then be used for:

- Comparisons with subway register counts. Exit counts should be treated as a lowerbound, since not all exiting passengers are counted.
- Checking the balance between entrance and exit total for a station, since usually they should be close.

For bus transactions, the Ride Check data is an alternate source of information. Counts of boardings, alightings and overall load are provided for each stop along each bus trip. Only a few routes were checked during our study period. These included the M31 bus on April 27 and the Q42 bus on April 28. We imported the available Ride Check data and successfully matched it to the STIF scheduled trips. We then tabulated the bus transactions by trip stop, so that the Ride Check data could be compared against our located transactions.

Most NYCT bus routes were not profiled during our study period. Results, up to a few years old, are available for most routes, aggregated to five time periods: 12am-6am, 6am-10am, 10am-3pm, 3pm-7pm, and 7pm-12am. Where the schedule patterns had not changed since the data collection, we were able in general to successfully match the stops with our STIF pattern stops. Given the incompleteness of this matching, we decided instead to tabulate our transactions to the stops along each TransCAD bus route, using the five time periods. We only tabulated those transactions that had TransCAD stop IDs assigned for both boarding and alighting. Transactions with only a DOT box number were often usable by tabulating to the adjacent stop if it had a stop ID. We then computed the load at each stop and indicated those stops with the maximum load value for a route during a time period. We observed a small number of negative loads, due to the assignment of transactions to time periods. This disappeared when we aggregate to a 24 hour period.

In order to compare the results from the automated location procedure with some known trips, we had our subcontractor, Howard/Stein-Hudson Associates (HSH), purchase some MetroCards and take ten predetermined tours, logging the boarding/alighting times and locations. We then compared our derived results to the actual trips. Tables 39-43 show five of the tours. The colored lines are from our procedure and alternate color based on the linked trips.

By in large, the results were quite promising given the quality of some of the datasets. The DOT bus trips were problematic, which was not surprising given the difficulty in identifying the scheduled trips to the location codes and route system. A couple of subway journeys used different pairs of trains to get between the same pair of stations. There were two EU65 transactions that were not present in the input dataset; it is unclear what caused this, since we do not think that the HSH employees jumped the turnstile.

Table 39: Tour 3, 4/21, Serial 0963940968

	Boarding Location	*Dep*	*Mode*	*Route*	*Alighting Location*	*Arr*
	28 St	9:06	Subway	1	Times Sq-42 St	9:10
1	Times Square	9:18	Subway	R	Bay Ridge 86th St / 4th Ave	10:11
	Times Sq-42 St	9:16	Subway	R	86 St	10:04
2	Bay Ridge 86th St/4th Ave	10:13	Bus - NYCT	S53	Castle Corners / Clove Rd / Victory Blvd	10:50
		10:06	Bus - NYCT	S53	4 Av & 90 St	10:08
3	Sunnyside / Clove Rd / Victory Blvd	11:14	Bus - NYCT	S61	St. George Ferry Terminal	11:26
	Bay St & Slosson Ter	11:12	Bus - NYCT	S66	ST GEORGE & FY	11:14
4	St George Ferry Terminal	11:32	SIR	SIR	Grassmere/Clove Rd	11:48
	St George	11:24	SIR	SIR	Grasmere	11:39
5	Grassmere / Clove Rd / Hylan Blvd	11:50	Bus - NYCT	S53	Bay Ridge 86th St / 4th Ave	12:05
	Clove Rd & Grasmere Station	11:48	Bus - NYCT	S53	86TH ST & 5TH AV	12:04
6	Bay Ridge 86th St / 4th Ave	12:11	Bus - NYCT	B64	Coney Island / Stilwell Ave / Mermaid Ave	12:35
	86 St & 5 Av	12:06	Bus - NYCT	B64	Stillwell Av & Mermaid Av	12:33
7	Coney Island / Stilwell Ave / Surf Ave	12:39	Bus - NYCT	B68	Brighton Beach Ave / Coney Island Ave	12:54
	Stillwell Av & Surf Av	12:36	Bus - NYCT	B68	Brighton Beach Av & Brighton 6 St	12:51
8	Brighton Beach Ave / Coney Island Ave	12:57	Subway	Q	Kings Highway / E 16th Street / Quentin	13:04
	Brighton Beach	12:48	Subway		Newkirk Av	12:58
9	Kings Highway / E 16th Street / Quentin	13:07	Bus - Cmd	B100	Fillmore / Flatbush	13:17
		13:00	Bus - Cmd			
10	Fillmore/Flatbush	13:27	Bus - NYCT	B41	Downtown Brooklyn / Cadman Pz / Tillary St	14:10
	FLATBUSH AV & AV N	13:18	Bus - NYCT	B41	Cadman Plz W & Johnson St	
11	High St / Brooklyn Bridge	14:20	Subway	A/C	34th St Penn Station	14:36
	High St	14:12	Subway	A	42 St-Port Authority Bus Terminal	14:32
	Times Sq-42 St	14:41	Subway	1	28 St	14:43

Table 40: Tour 5, 4/29, Serial 0959419907

	Boarding Location	*Dep*	*Mode*	*Route*	*Alighting Location*	*Arr*
1	34th St / 10th Ave	8:58	Walk		39th St / Madison Ave	9:14
2	Madison Ave / 39 St SE Corner	9:30	Exp - Liberty	BxM11	Bronx Zoo / Entrance	10:08
		9:30	Exp - Liberty			
2A	Bronx Zoo / Entrance		Walk		Bronx Park East	
3	Bronx Park East	10:35	Subway	2	149th St / Grand Concourse	10:49
	Pelham Pkwy	10:18	Subway	2	149 St-Grand Concourse	10:41
4	149th St / Grand Concourse	11:04	Subway	5	125th St / Lexington	11:09
	149 St-Grand Concourse	10:48	Subway	4	86 St	10:56
5	125th St / Lexington	11:25	Bus - NYCT	M60	LGA / Delta Terminal	11:50
		11:18	Bus - NYCT			
6	LGA / Delta Terminal	12:25	Bus - NYCT	Q48	Flushing - Main St / Roosevelt Ave	12:56
		12:18	Bus - NYCT			
7	Flushing - Main St / Roosevelt Ave	13:10	Subway	7	Queensboro Plaza	13:32
	Flushing-Main St	12:54	Subway	7	Junction Blvd	1:04
8	Queensboro Pz N / Crescent St	13:58	Bus - Qns S	Q102	Roosevelt Island - Coler / Gold Water	14:18
		1:54	Bus - Qns S			
9	Roosevelt Island	14:30	Tramway	RIT	59th St / 2nd Ave	14:35
	Roosevelt Island	14:18	Tramway	RIT	Manhattan-2 Av	15:23
10	Lexington / 59th St	14:50	Subway	N	42nd St Times Square	15:00
	59 St	14:42	Subway	5	Grand Central-42 St	14:46

Table 41: Tour 8, 4/27, Serial 0955395901

	Boarding Location	*Dep*	*Mode*	*Route*	*Alighting Location*	*Arr*
1	Riverdale - W. 231st St / Broadway	8:54	Subway	1	59th St / Columbus Circle	9:24
	231 St	8:48	Subway	1	96 St	9:11
2	59th St / Columbus Circle	9:29	Subway	C	Borough Hall - Jay St	9:51
	96 St		Subway	2	Clark St	9:40
3	Jay St / Livingston St	9:55	Bus - NYCT	B41	Church Ave / Flatbush Ave	10:23
	Cadman Plz W & Johnson St	9:54	Bus - NYCT	B41	Flatbush Av & Church Av	10:23
4	Church Ave / Flatbush Ave	10:38	Bus - NYCT	B35	Church Ave / McDonald Ave	10:54
	Church Av & Flatbush Av	10:30	Bus - NYCT	B35	Church Av & New York Av	10:31
5	Church Ave / McDonald Ave	11:00	Subway	F	Borough Hall - Jay St	11:15
	Church Av	10:54	Subway	F	Broadway-Lafayette St	11:24
6	Borough Hall - Jay St	11:16	Subway	A	59th St / Columbus Circle	11:38
	Bleecker St	11:30	Subway	6	28 St	11:36
7	59th St / Columbus Circle	11:42	Bus - NYCT	1	Riverdale - W. 231st St / Broadway	12:12
	NO EU65 Transaction					

Table 42: Tour 9, 4/28, Serial 0959419908

	Boarding Location	*Dep*	*Mode*	*Route*	*Alighting Location*	*Arr*
1	W 4th St / 6th Ave	8:24	Subway	A	Borough Hall - Jay St / Willoughby St	8:35
	W 4 St	8:18	Subway	A	High St	8:28
2	Borough Hall - Jay St / Willoughby St	8:41	Bus - NYCT	B67	Park Slope - Union St / 7th Ave	8:53
	Jay St & Nassau St	8:36	Bus - NYCT	B67	7 Av & Union St	8:56
3	Park Slope - Union St / 7th Ave	9:13	Bus - NYCT	B71	Cobble Hill - Smith St / Union St	9:27
	Union St & 7 Av	9:12	Bus - NYCT	B71	Smith St & 2 St	9:21
4	Cobble Hill - Smith St / Union St	9:41	Bus - NYCT	B75	Downtown Brooklyn - Jay St / Willoughby St	9:54
		9:36	Bus - NYCT			
5	Downtown Brooklyn - Jay St/Willoughby St	9:57	Subway	A	34th Street / Penn Station	10:14
	Jay St-Borough Hall	9:54	Subway	A	W 4 St	10:05

Table 43: Tour 10, 4/29, Serial 0963940968

	Boarding Location	*Dep*	*Mode*	*Route*	*Alighting Location*	*Arr*
1	34th St / 10th Ave	13:12	Bus - NYCT	M34	34th St / 6th Ave	13:30
	W 34 St & 8 Av	13:06	Bus - NYCT	M34	W 34 St & Broadway	13:10
2	34th St / 6th Ave	13:37	Subway	F	Avenue I / MacDonald Ave	14:15
	34 St-Herald Sq	13:30	Subway	F	Avenue I	14:05
3	Avenue I / MacDonald Ave	14:18	Bus - NYCT	B11	Flatbush Ave / Nostrand Ave	14:42
	Av I & McDonald Av	14:18	Bus - NYCT	B11	Nostrand Av & Flatbush Av	14:43
4	Flatbush Ave / Nostrand Ave	14:49	Subway	2	Hoyt St	15:10
	NO EU65 Transaction		Subway			
5	Hoyt St / Fulton St	15:10	Walk		Hoyt St / Fulton St	15:50
6	Hoyt St	15:56	Subway	2 or 3	Wall Street	16:15
	Hoyt St	15:48	Subway	2	Wall St	15:57
7	Wall Street	16:15	Walk		Wall Street	16:40
8	Wall Street	16:55	Subway	2 or 3	34th St / 7th Ave	17:15
	Wall St	16:54	Subway	2	Fulton St	16:56
	Chambers St	17:02	Subway	N	34 St-Penn Station	17:14

We also had HSH and NYCT ask for volunteers, who logged many of their normal commuting trips. Table 44 shows one such log.

Table 44: Commuter Log, Serial 0894670868

	Boarding Location	*Dep*	*Mode*	*Route*	*Alighting Location*	*Arr*
4/21	97th and Central Park West IND Station	8:40	Subway	C	35th and 8th Ave. IND Station	8:59
	96 St	8:36	Subway	C	59 St-Columbus Circle	8:42
	59 St-Columbus Circle	8:50	Subway	1	66 St-Lincoln Center	8:51
	66 St-Lincoln Center	23:12	Subway	1	59 St-Columbus Circle	23:19
	59 St-Columbus Circle	23:26	Subway	A	96 St	23:31
4/27	97th and Central Park West IND Station	8:16	Subway	C	35th and 8th Ave. IND Station	8:29
	96 St	8:12	Subway	B	59 St-Columbus Circle	8:19
	59 St-Columbus Circle	8:24	Subway	A	34 St-Penn Station	8:28
	35th and 8th Ave. IND Station	17:57	Subway	C	97th and Central Park West IND Station	18:10
	34 St-Penn Station	17:54	Subway	C	96 St	18:15
4/28	97th and Central Park West IND Station	8:55	Subway	C	35th and 8th Ave. IND Station	9:09
	96 St	8:42	Subway	C	34 St-Penn Station	8:59
	35th and 8th Ave. IND Station	17:03	Subway	C	72nd and Central Park West IND Station	17:11
	34 St-Penn Station	17:00	Subway	E	42 St-Port Authority Bus Terminal	17:02
	Times Sq-42 St	17:08	Subway	2	72 St	17:12
	72nd and Broadway IRT Station	19:00	Subway	9	96th St. and Broadway IRT station, NY, NY	19:05
	72 St	18:54	Subway	1	59 St-Columbus Circle	19:09
	59 St-Columbus Circle	19:08	Subway	C	96 St	19:13
4/29	97th and Central Park West IND Station	8:26	Subway	C	35th and 8th Ave. IND Station	8:38
	96 St	8:24	Subway	C	34 St-Penn Station	8:38
	34 St-Penn Station	16:12	Subway	C	96 St	16:24
5/1	96 St	7:12	Subway	1	50 St	19:20
	50 St	22:03	Subway	1	96 St	22:43

A broader and equally important aspect of the validation was the examination of the results generated for particular known locations familiar to NYCT expert staff. Throughout the project, NYCT staff scrutinized the results of the evolving transit trip O-D database. Load profiles for various test routes were repeatedly examined and corrective measures were taken to improve the results.

In the later phases of the project, a series of transit assignments were run. These included stop to stop assignments and subsequently zone to zone assignments. We computed zone to zone trip matrices that were then compared to previous trip matrices used for forecasting and the 2000 Census journey-to-work flows by transit. Bus and rail and combined matrices were also used as inputs to a Transit Assignment procedure in TransCAD to assess the quality of the zone to zone matrices. Output from the assignments was helpful in identifying needed corrections in the methodology for creating the zone to zone tables.

Query Software

Powerful query software was created for this project that allows almost any conceivable query to be answered; the software works in two steps: trip/leg selection and output creation. Queries can be made on either the linked trips table or the unlinked legs table. The *Query Builder* step defines a query, combining one or more primitives that conceptually select a set of trips/legs (see Figure 6). The query can require that either any or all of the primitives be matched. For linked trips, the primitives include selecting by the mode, route, or pattern of a particular leg in the trip sequence; by the origin and/or destination of the trip by specifying stops, zones, Census Tracts or Boroughs; by the inclusion of a particular type of transfer between modes within the trip; or by a general SQL query. For unlinked legs, the primitives include selecting by the mode, route or pattern of the leg; by the origin and/or destination of the leg by specifying stops, zones, Census Tracts or Boroughs; by specifying the mode of either the proceeding or following leg used as part of a linked trip; by a general SQL query; or by the position of the leg within its linked trip. Queries can be saved for later reuse.

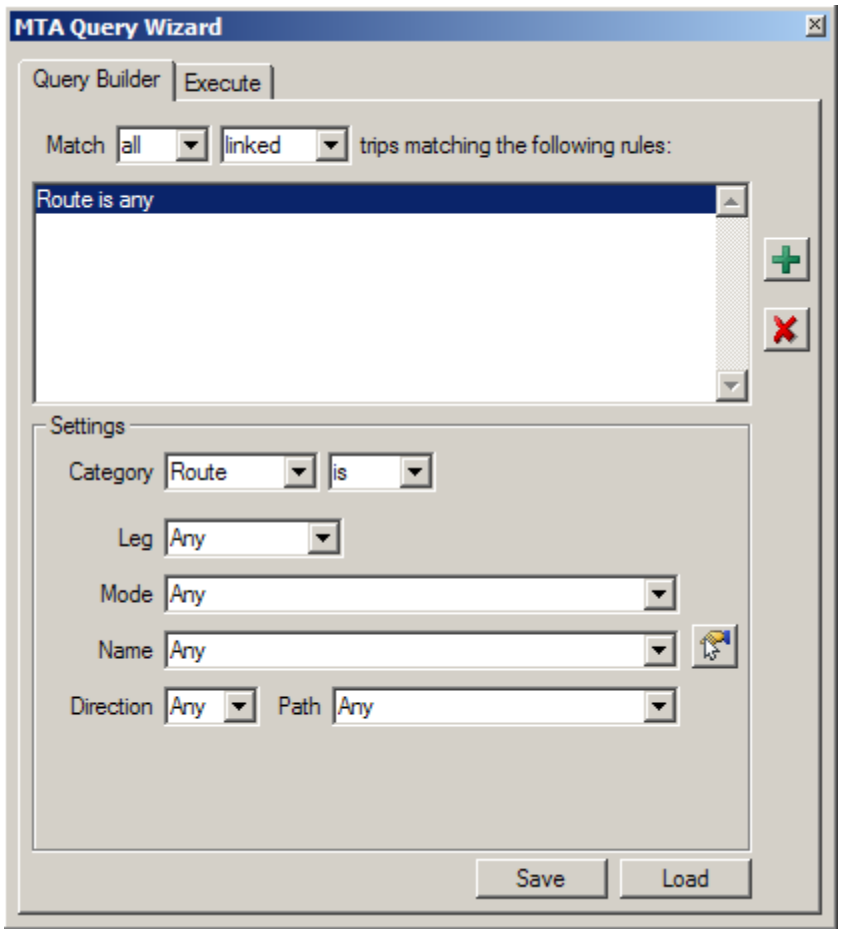

Figure 6: Initial Query Toolbox

The *Execute* step specifies the day to use for the query, selected from the list of available days (see Figure 7). The trips/legs can be further restricted based on a time period and/or an existing TransCAD selection set of linked trips. The choices for the output include reports (see Figure 9), maps (see Figure 8), O-D matrices or a TransCAD selection set that can be used to create external tables/spreadsheets. The reports can be summarized by arrival time, departure time, mode or route of a particular leg, origin, destination, and/or O-D pair. A ridership report by either route pattern or route segment can also be produced. Maps can depict the origins and/or destinations of the trips/legs selected. Maps can also include a scaled theme depicting the ridership by street/track segment. The O-D matrices summarize the trips/legs by stop, zone, Census Tract or Borough. Besides creating export tables, the selection set can be used for general analysis in TransCAD or for examining individual linked trips in a customized trip browser that depicts each leg of the linked trip on the map.

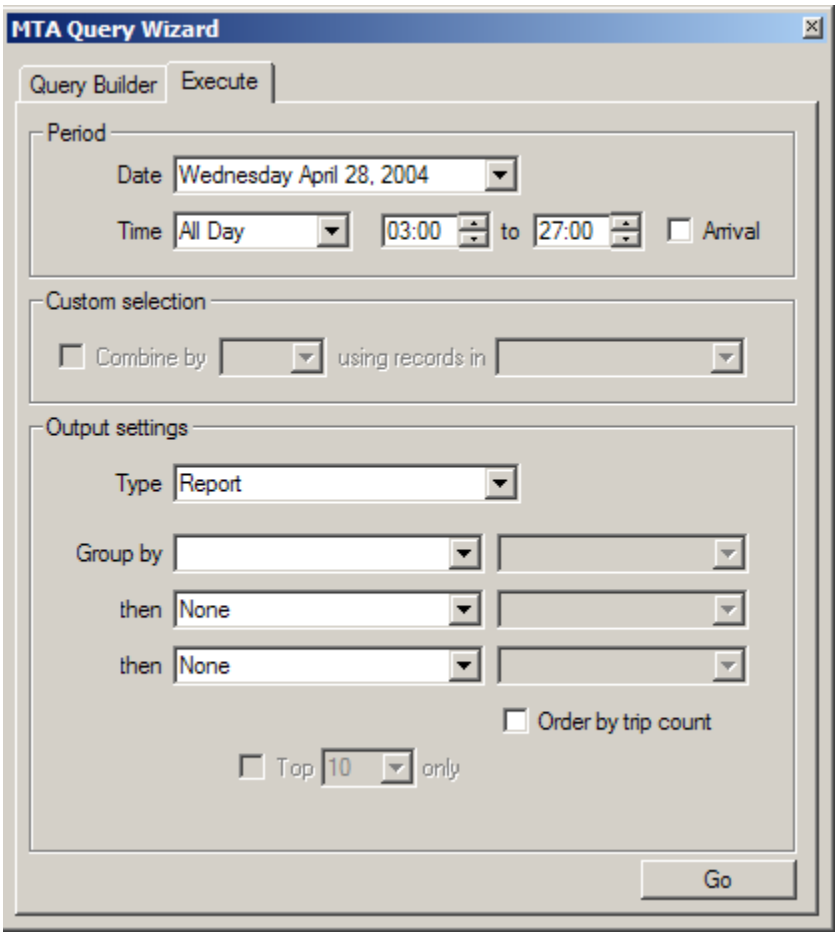

Figure 7: Initial Execute Tab

See the *User's Guide for the MetroCard Query Software* for complete details on how to install and use the Query Software.

Figure 8: Subway Ridership Map

Route is Subway

Wednesday April 28, 2004, All Day

Route 1			
Interval	AB Riders	BA Riders	Total Riders
South Ferry -- n/o South Ferry	4,354	0	4,354
n/o South Ferry -- Rector St	4,354	1,839	6,193
Rector St -- Chambers St	5,021	3,437	8,458
Chambers St -- Franklin St	5,692	3,854	9,546
Franklin St -- Canal St	5,548	4,948	10,496
Canal St -- Houston St	5,499	6,167	11,666
Houston St -- Christopher St-Sheridan Sq	5,229	9,388	14,617
Christopher St-Sheridan Sq -- 14 St	6,066	9,947	16,013
14 St -- 18 St	9,465	10,081	19,546
18 St -- 23 St	9,302	11,247	20,549
23 St -- 28 St	9,453	12,945	22,398
28 St -- 34 St-Penn Station	9,344	15,225	24,569
34 St-Penn Station -- Times Sq-42 St	14,377	15,603	29,980
Times Sq-42 St -- 50 St	17,542	19,080	36,622
50 St -- 59 St-Columbus Circle	13,685	21,994	35,679
59 St-Columbus Circle -- 66 St-Lincoln Center	13,636	24,097	37,733
66 St-Lincoln Center -- 72 St	10,481	24,190	34,671
72 St -- 79 St	10,531	22,931	33,462

Figure 9: Subway Ridership Report

Maintenance

The processing software written for the project makes use of a consistent collection of schedules, TransCAD route system and other input tables from the study period in April 2004. This section discusses the changes required in order to use the software with data from other time periods.

The primary inputs are the EU65 MetroCard transactions and the AFC bus trip table. As long as neither file format changes, we suspect the software should be able to process the data, but the accuracy will degrade as the current schedules diverge from the 2004 schedules. The private DOT bus routes are now operated by MTA bus and we suspect that their codes have radically changed and probably not handled correctly.

In order to process other periods correctly, the following updates are required:

1. The TransCAD route system should be updated to reflect as accurately as possible all the route patterns for the time period. The subway routes need to match the RTIF schedules exactly so that the schedule based shortest path procedure can be used. The same is true for the SIF and SIR. The redundant attributes should be consistent between route, stops, physical stops and/or intersections layers. The DOT box numbers for bus stops should match the schedule. Each zone centroid needs to be connected to the street intersections.

2. The bus schedules need to be imported into TransCAD for weekdays, Saturday and Sunday. Currently there are three files for each time period: NYCT, LIB and DOT. Every schedule stop needs to be assigned a DOT box number and if possible geographic coordinates. The schedule trips need to be matched against the route system and the sequence of stops adjusted if the schedule and route system do not agree.

3. The subway (RTIF), SIF, and SIR schedules need to be imported into TransCAD and converted into the SSP format.

4. The tatmaster file should be updated to reflect the current set of location and destination sign codes. This is manually used to derive the pattern table.

5. The list of open subway stations may need to be updated.

6. The list of bus stop locations (DOT boxes) needs to be updated.

7. The processing logic needs to be checked and possibly changed to handle the MTA bus routes. Westchester's Bee Line could also be handled if necessary.

8. We believe PATH is going to accept MetroCard at more stations and the processing logic should be check to verify this does not cause any problems.

9. The weekday estimates of commuter rail/bus traffic should be updated.

Some of the above steps could be further complicated if the format or composition of any of the inputs changed from those we received in 2004. It is possible that switching to HASTUS may change the contents of the bus schedules.

Concluding Remarks

As should be evident from this report, the processing procedure developed to generate the O-D trip database is highly complex, involves numerous intermediate steps, and many assumptions and some sampling approximations. Undoubtedly, there are errors in the trip tables produced especially for bus trips. Nevertheless, the portrait of NYCT system utilization is probably the most accurate that has ever been available and it has many potential applications.

Users of the data should exercise caution in working with the data as some corrections may needed for specific purposes. Alternatively, some of the positional errors in bus boardings can be probably be lessened by aggregating the data geographically which can easily be done in TransCAD. For some applications, supplemental data can be used to enhance and further correct the output from this system.

There are many ways the data development approach could be improved. The following improvements would improve the accuracy of the results that could be obtained from MetroCard transactions. Some of them would also allow simpler algorithms to be employed for the processing.

1. Improve the accuracy of recorded MetroCard transactions to a minute, a second or better. A bus can travel a significant distance in six minutes.

2. Preserve the orderings of the MetroCard bus boardings.

3. Improve the route system to the point where it matches the schedules exactly and also has accurate geographic locations for all the stops.

4. Improve the bus trip logging system to allow an easier recovery of trip records. This could include having the drivers enter sign codes more consistently. Even better would be to include a GPS-based system to provide bus locations.

Ironically, although there are no technical obstacles, there is more GIS work to do in representing all service patterns that are in use in New York City. Because of the processing method, full representation of routes and schedules in a consistent, GIS-based fashion will be helpful in carrying this system forward.

The data provide the opportunity to perform much more sophisticated types of forecasting utilization of new services by making it possible to use dynamic (i.e., time dependent)

methods of transit assignment. Further calibration of the transit assignment procedure could work hand-in-hand in making fuller use of the data generated.

Lastly, a rider survey that can be matched to a MetroCard ID and set of data transactions would provide valuable additional data for travel demand modeling especially if route choice and demographics are accurately captured. A survey would also provide further insight into the quality of the procedures employed and might also suggest some additional enhancements or modifications to be made in the future.

References

[BNRS02] J. J. Barry, R. Newhouser, A. Rahbee and S. Sayeda. Origin and Destination Estimation in New York City with Automated Fare System Data. *Transportation Research Record* 1817:183-187, 2002.

[RC02] A. Rahbee and D. Czerwinski. Using Entry-Only Automatic Fare Collection Data to Estimate Rail Transit Passenger Flows at CTA. In *Proc. 2002 Transport Chicago Conf.*, 2002.