SRF Consulting Group, Inc. | Engineers Planners Designers

SRF No. 0117416

DRAFT TECHNICAL MEMORANDUM

TO: Mark Filipi, Metropolitan Council
Jonathan Ehrlich, Metropolitan Council

FROM: Steve Wilson, Principal
Alex Fox, Senior Engineer
Nathan Koster, Senior Planner

DATE: December 30, 2011

SUBJECT: TWIN CITIES 2010/2030 REGIONAL MODEL TRANSIT NETWORK DEVELOPMENT

INTRODUCTION

This memorandum documents the methodology used to integrate the Metropolitan Council's travel demand model transit networks with the geodatabase structure and conflation of the highway network completed in 2010. The transit network structure is created as a Citilabs CUBE geodatabase, with options to use as either TRNBUILD or PT format. This memorandum includes a description of the data used, modifications to the highway network geodatabase, generation of route frequency information, general quality control/quality assurance measures and network validation.

The attached memorandum includes information on the TRNBUILD to Public Transport(PT) conversion, the PT skimming and assignment application and the geodatabase structure (Appendix B).

DATA REQUIREMENTS

The 2010 transit network was generated to be consistent with the Twin Cities regional transit system as of September 2010 for use with the 2010 Travel Behavior Inventory (TBI) model update. The update used the following datasets:

1. September 2010 pattern GIS shapefile
2. September 2010 Weekly transit vehicle trip data
3. September 2010 Automatic Passenger Count (APC) data
4. September 2010 park and ride GIS shapefile
5. Twin Cities Regional Travel Demand Model network geodatabase as of April 2011

The pattern shapefile provides the spatially correct line work for each transit pattern in the system. A transit pattern represents a distinct transit route permutation with a specific frequency. Each transit pattern was converted into one transit line in the transit model. Transit routes are composed of multiple transit patterns serving a similar travel market. Vehicle trip data is used to generate route frequencies.

METHODOLOGY

The transit network update included modifications to the transit line file, transit support links and modifications to the regional highway network. The methodology for developing the transit line file includes both geoprocessing in ArcGIS and data formatting and integration.

Pattern Shapefile Modification

Metro Transit provided formatted route pattern shapefiles for each pattern. ArcGIS geoprocessing was used to automate the generation of the transit line file as opposed to coding them individually. Inputs to this process included the highway network node feature class and a well formatted pattern shapefile. Both inputs required modifications and systematic quality control checks to produce a high quality network.

The pattern shapefile needed to include the following:

1. A unique record for each route pattern that directly corresponds to trip schedule data;
2. Each route pattern must have one continuous line with no spatial discontinuities;
3. Each route pattern must have sequentially ordered parts.

Metro Transit provided a pattern shapefile consistent with trip schedule data; however, some pattern identifiers were not unique across routes. Figure 1 depicts an example of one pattern identifier that is included in two separate routes. Since the Metro Transit pattern identifiers are a concatenation of the pattern start and end time points, any pattern with the same start and end time point will have redundant identifiers. Additional processing of the pattern file was needed to generate a unique route pattern identifier. Consequently, an underscore route number was added to the identifier for each pattern in order to generate a unique route pattern identifier. Figure 2 depicts the pattern from Figure 1 split into separate patterns for each route (Route 94 and Route 3).

Figure 1-Example Metro Transit Pattern Redundancy

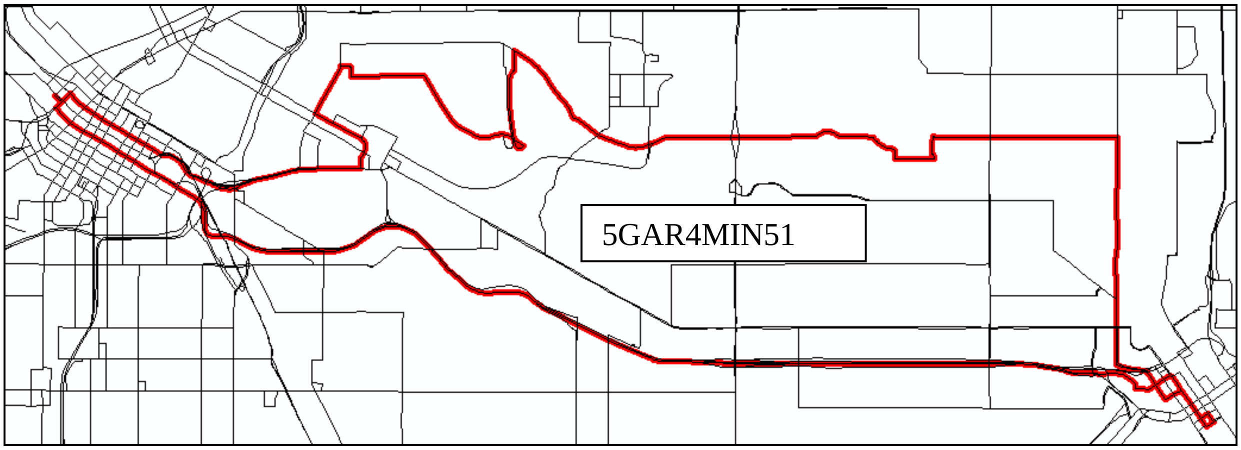
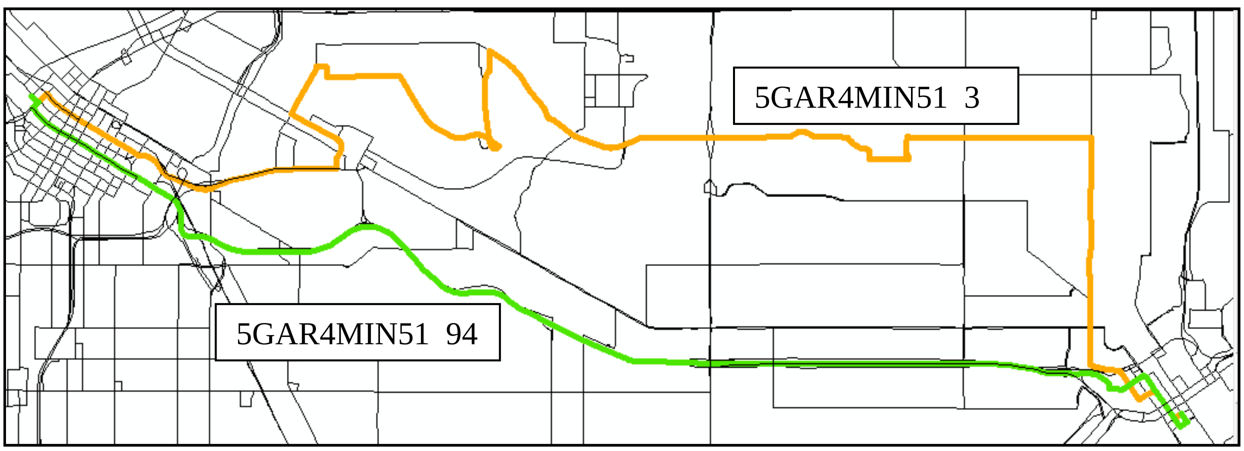


Figure 2- Split Transit Pattern IDs



Spatial discontinuities within the patterns were identified and rectified in ArcGIS (Figure 3). Quality checks were conducted using the CUBE highway network builder to identify these locations. Figure 4 provides an example of the initial and current iteration of the CUBE network builder QA/QC process.

Figure 3-Pattern Discontinuity (Route 698)

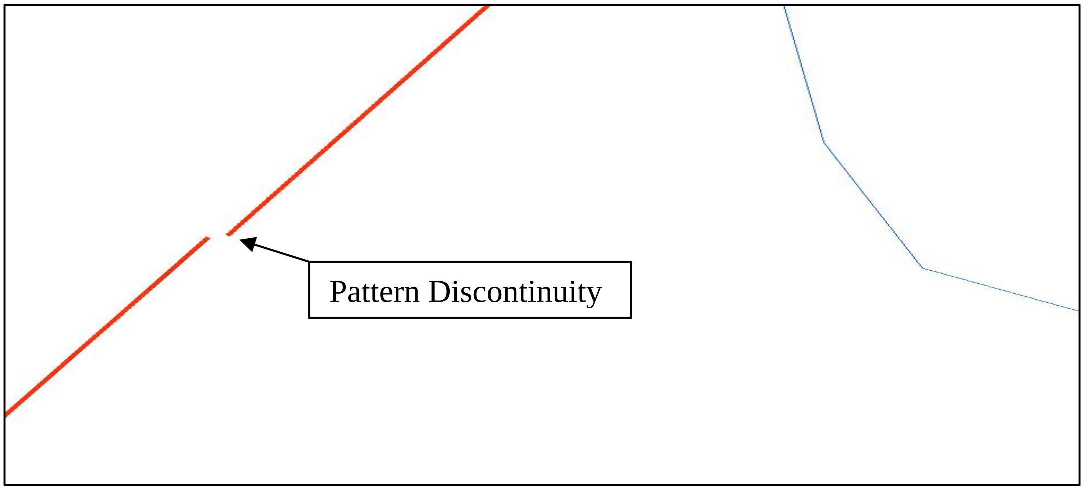
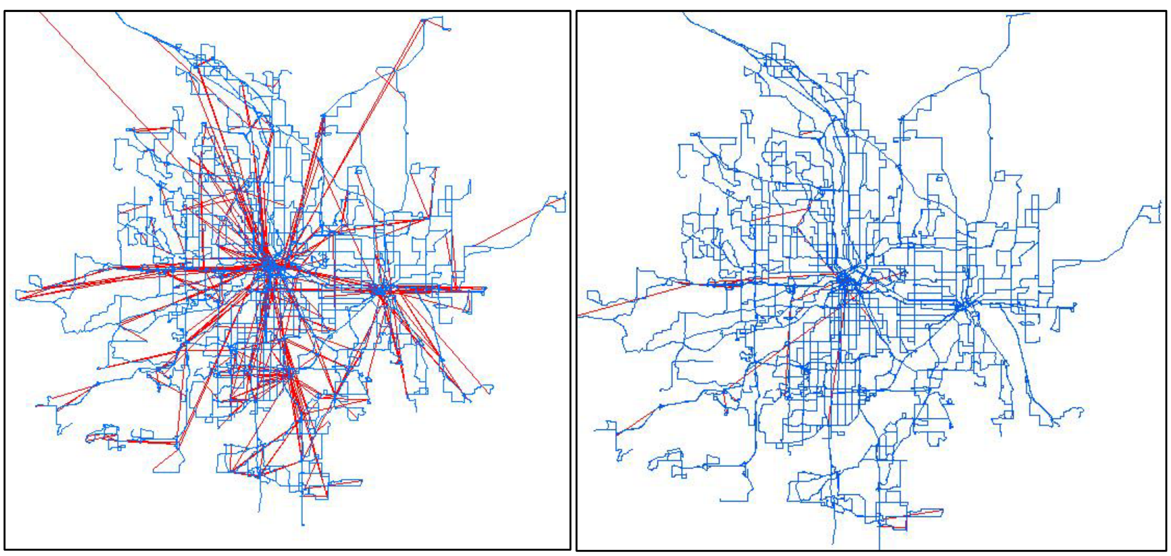
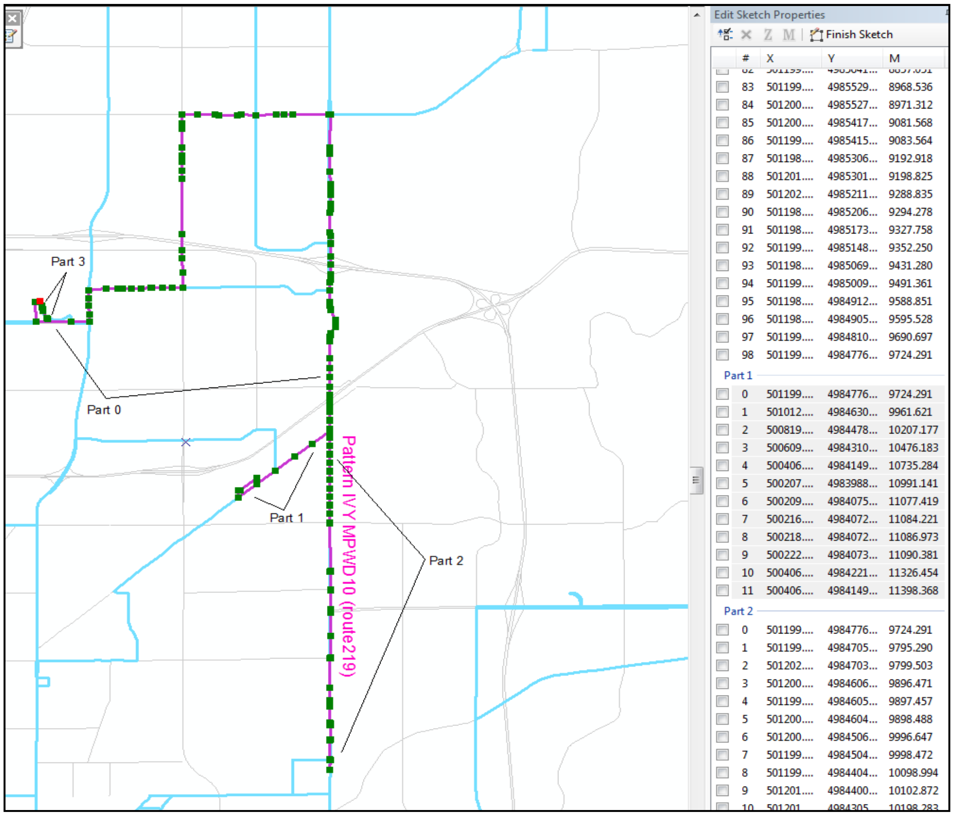


Figure 4-Pattern Discontinuity QAQC (Initial Iteration-Left, Revised Iteration-Right)



Another issue with the pattern files was the sequencing of some pattern parts resulted in the loss of spatial conflation (Figure 5). Some of the mis-ordering is legitimately related to breaks at park and ride locations and other pattern discontinuities, but most are due to the methodology used to generate initially generate the pattern file. The result is an inability to cleanly generate a linearly referenced node sequence for the route. The problem locations were identified iteratively and the out-of-order parts were reordered or consolidated.

Figure 5- Out-of-Order Pattern Segments



Modifications were made to several patterns in order to improve the geoprocessing conversion process by identifying and correcting overlapping pattern segments. Figure 6 depicts one overlapping pattern segment resulting from the pattern traveling in and out of the Rosedale Mall. The geoprocessing implemented does not include a node multiple times in a single pattern node sequence. Patterns with significant overlapping segments were identified for editing in CUBE. Overlapping segments that were not significant were removed from the pattern shape file prior to geoprocessing.

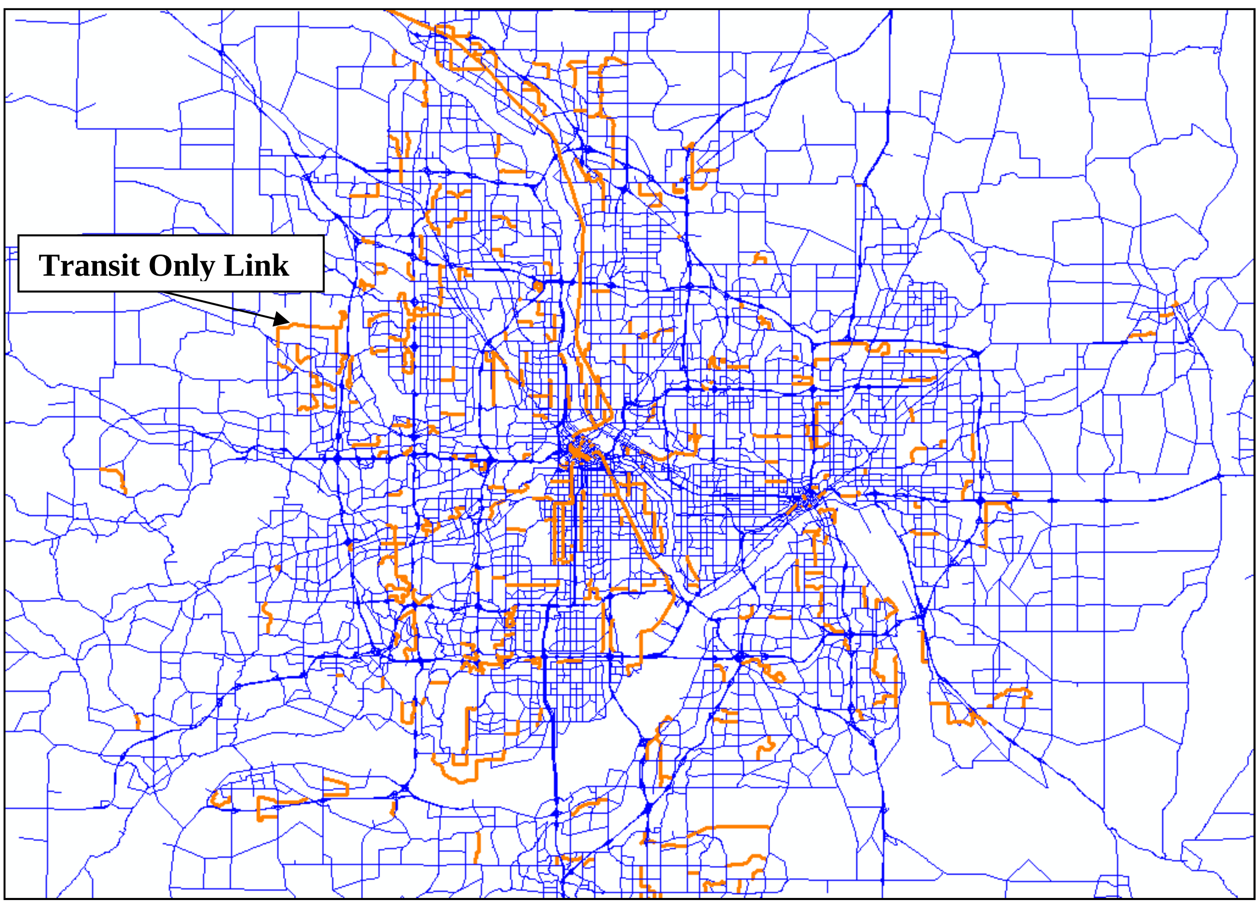
Figure 6- Overlapping Pattern Segment

Highway Network Modification

To best make use of the spatial correctness and relationships within the geodatabase format the overall transportation link structure of the geodatabase was modified to include most additional transit patterns and route segments not included in the original highway network feature class. Prior to making edits to the pattern shapefile, adjustments were also made to the highway network. Transit-only network links, in orange, were added to provide more detailed connectivity and improved spatial depiction of the transit system (Figure 7). These links are deleted in the network preparation script for all highway model processes.

Some additional network edits were performed and errors corrected. Some network nodes were slightly shifted to better align with the transit pattern shapefile and improve the linear referencing process. Additionally, network errors near both Hemlock Lane and Maple Grove Parkway in Maple grove were fixed to accommodate existing and future transit routes in the area.

Figure 7 –Transit Only Network Links



Sensitivity tests were conducted on the highway network to determine the impact of the highway network modifications on model traffic volumes. It was determined that the impact on the overall assignment process was minimal. Table 1 depicts the minimal affect on the loaded model volumes by volume group. The total regional percent difference of daily model volumes is 0.3%.

Table 1 –Percent Difference of Loaded Daily Traffic Volumes

Volume Group	Network Update vs. Base Network (Shape length used for distance)
0-1,000	0.9 %
1,000-10,000	0.5 %
10,000-20,000	0.4 %
20,000+	0.1 %
Total	**0.3 %**

Geoprocessing

ArcGIS geoprocessing was conducted in two steps. First, the patterns were converted to routes which define a definitive beginning and end point for each pattern in the shapefile. Second, the locate features along routes process was used to convert the route file from step one to an event table (Table 2).

Table 2 –Event Table Example

Pattern ID	Measure	Node
1022WA2A00	1.9229	6299
1022WA2A00	2.3070	18017
1022WA2A00	2.3596	18010
1022WA2A00	2.4047	6246
1022WA2A00	2.6089	6307
1022WA2A00	3.1016	6304
1022WA2A00	3.6001	6311
1022WA2A00	3.8512	6310
1022WA2A00	4.0964	6314
1022WA2A00	4.3737	6315

The event table includes a node sequence from the start to the end of each pattern. This pattern specific node sequence is the basis for the transit network. The pattern ID represents the unique pattern identifier for each route. The measure field is a linear distance measurement along the path from the beginning of the pattern providing a linear referencing index. It provides sequencing for the network nodes provided in the node field.

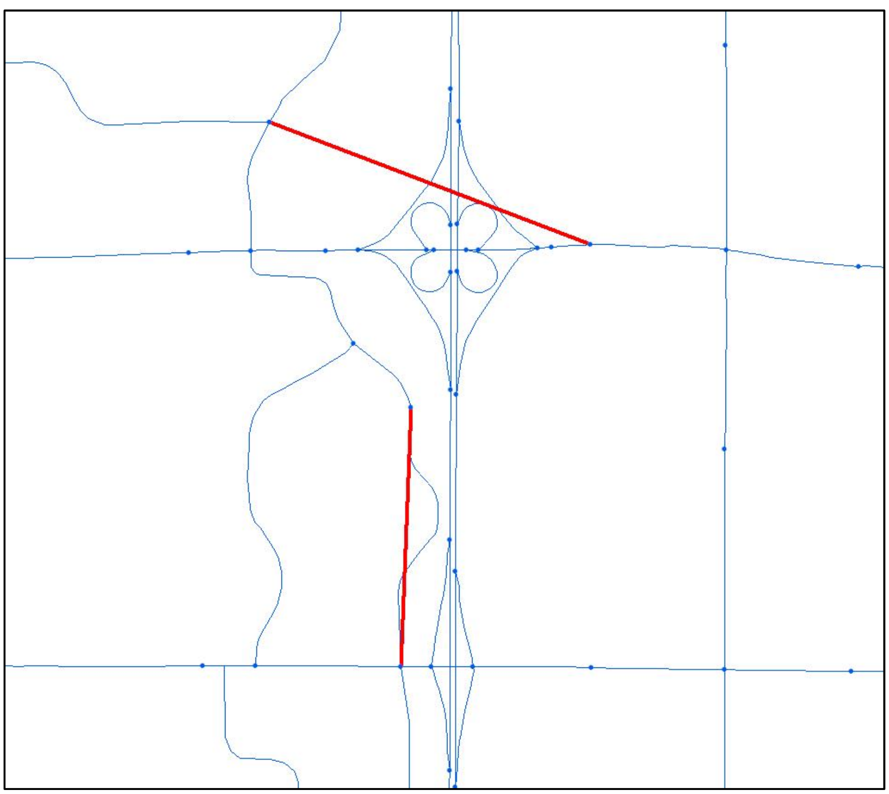
Pattern direction was reviewed and node sequences were reversed as necessary. This review was needed because the beginning and end of the patterns were generated randomly and not necessarily based on the actual direction of the route. One way roadway segments such as freeways, ramps and one-way arterials were used to identify these wrong way patterns.

Next, the nodes sequenced for each pattern were analyzed on the basis of stop and non-stop. APC data was spatially joined to a shapefile of potential network stop nodes by pattern. Potential network nodes included any node not a centroid or connected to a network freeway link (with assignment group 1,2,8,10,13,14). This data was joined to the pattern node sequences and used to identify the nodes as stop and non-stop. Stop nodes for patterns that did not have APC data provided by Metro Transit were manually reviewed.

Node Sequencing QA/QC

After completion of the ArcGIS geoprocessing, a systematic check of the node sequencing was performed. A node sequence for each unique transit pattern was compiled into a database. The node sequences contained all unique A-B node pairs, which were integrated into the highway network, as network links, to identify spatial differences between the transit and highway network. Differences between the transit and highway network were flagged and corrected in the transit node sequencing (Figure 8).

Figure 8 – Node Sequencing Deviations



Pattern Data Integration

Once the node sequences were established, headway, mode, route number and pattern ID data were integrated into the database. The transit line file was developed for all regional transit route patterns serving the AM (6 a.m. – 9 a.m.) and midday (9 a.m. – 3 p.m.) time periods. Trips were assigned to the time periods based on the average start and end time. Patterns with single trips in a time period were reviewed for reasonability and adjustments were made to account for a.m.-only patterns unnecessarily overlapping into the midday time period. Transit vehicle trips per pattern were aggregated and converted into headway for use in the model. Table 3 includes the trip to headway conversion table assumed.

Table 3 –Trips to Headway Conversion

AM Trips 6 a.m. - 9 a.m.	AM Modeled Headway (minutes)	Midday Trips 9 a.m. - 3 p.m.	Midday Modeled Headway (minutes)
1	180	1	360
2	90	2	180
3	60	3	120
4	45	4	90
5	36	5	72
6	30	6	60
7	26	7	51
8	22.5	8	45
9	20	9	40
10	18	10	36
11	16	11	33
12	15	12	30
13	14	13	28
16	11	14	26
19	9.5	15	24
22	8	16	22.5
25	7	18	20
26	7	20	18
		24	15
		34	10.5
		35	10
		36	10
		44	8
		71	5
		72	5

TRANSIT NETWORK

Metro Transit had 218 a.m. and midday routes in service by September 2010. These routes were converted to 821 transit lines for the model by coding each route pattern and route direction as a separate line. Mode 3 dummy support links were not used since PT does not support this type of transit connectivity pattern. The mode structure of the model is as follows:

- Mode 1 – Walk Access Links (Non-Transit)
- Mode 2 – Drive Access Links (Non-Transit)
- Mode 4 – Downtown Walk Network Links (Non-Transit)
- Mode 5 – Urban Local (Transit)
- Mode 6 – Suburban Local (Transit)
- Mode 7 – Express (Transit)
- Mode 8 – Light Rail (Transit)
- Mode 9 – Commuter Rail (Transit)
- Mode 10 – Premium Bus (Transit)

Non-Transit Links

Mode 1 walk access support links were regenerated based on the updated transit network stop node structure. Mode 1 walk access links are two-directional connections from TAZs to transit stations with a maximum distance of 1 mile throughout the metropolitan area. Mode 4 walk access links represent downtown Minneapolis and St. Paul walk networks. This mode was introduced to better model trips in central business districts. The speed of the walk access links was set to 2.5 mph to reflect the effects of route circuitry and traffic signal delay in the downtown areas. This speed has been observed in GPS data collection efforts.

Mode 2 drive access support links were regenerated based on the new stop node structure and the existing and future planned park and ride system. Drive access links represent either connection from zones to transit stations or connection from zones to park-and-ride facilities. Regular drive support links are 1.5 mile long in the urban core, excluding downtown Minneapolis and Saint Paul, and are 2.5 miles long in the suburbs. Park-and-ride access links were coded by using the following assumptions:

Table 4 – Park and Ride Access Link Coding Assumptions

Park-and-Ride Lot Type	Lot Capacity	Driving Direction	Maximum Drive Time(1)
Small	<100	All directions	7 minutes
Medium	100-200	All directions	7 minutes
Large	>200	Away from CBD	7 minutes
Large	>200	Toward CBD	22.5 minutes

(1) Based on time period-weighted travel times developed in model MSA process

Transit Lines

The PT transit network lines are stored and edited in a geodatabase PT feature class; however, the non-transit support links and model parameter files are stored in text file format. The structures of the transit line and non-transit supportlink files are provided below.

- Transit Lines (fields: Name, Longname, Headway_1, Headway_2, Mode, Operator, UserA5, Node List)
- Walk Access Links (fields: Anode, Bnode, Mode, Distance, Cost, Speed, One-Way)
- Drive Access Links (fields: Anode, Bnode, Mode, Distance, Cost, Speed, One-Way)
- Downtown Walk Links (fields: Anode, Bnode, Mode, Distance, Cost, Speed, One-Way)

Three transit line identifiers are provided in the transit network. The *name* field is a unique line identifier that meets the maximum character requirements of 28 characters for PT (compared to 12 in TRNBUILD). In general, the *name* field includes route number, direction and end terminus of the route. The *UserA5* attribute is the route number. The *longname* attribute provides a modified metro transit pattern identifier. This attribute is unique and can be used to integrate additional metro transit pattern data as desired. The *longname* attribute provides a direct one to one correspondence for most of the transit lines with the "pat_id" field in the Metro Transit trips database. In some locations where the "pat_id" field was not unique, the *longname* attribute in the model was modified to include an underscore route number "_3".
For example:

- Name – Ex: 783_EB_BALW
- UserA5 – Ex: 783
- Longname – Ex: BALWMA1050
- Longname (Where value is not unique in Metro Transit trips database) – Ex: 5GAR4MIN52_3

See Appendix A for additional detail regarding attribute definitions.

Transit Speeds

Transit speeds and associated travel times were input into the transit model according to Table 5. The speeds are estimated based on a systematic analysis of route-level scheduled run times averaged across all routes and stratified by assignment group and area type. The transit network preparation program directly calculates transit travel time from assignment group and area type from a lookup table. Manually specified stop to stop transit travel time is input into the highway network under attribute T_MANTIME. A manual transit travel time is applied for existing and future fixed guideways.

Table 5 –Transit Speeds (Miles per Hour)

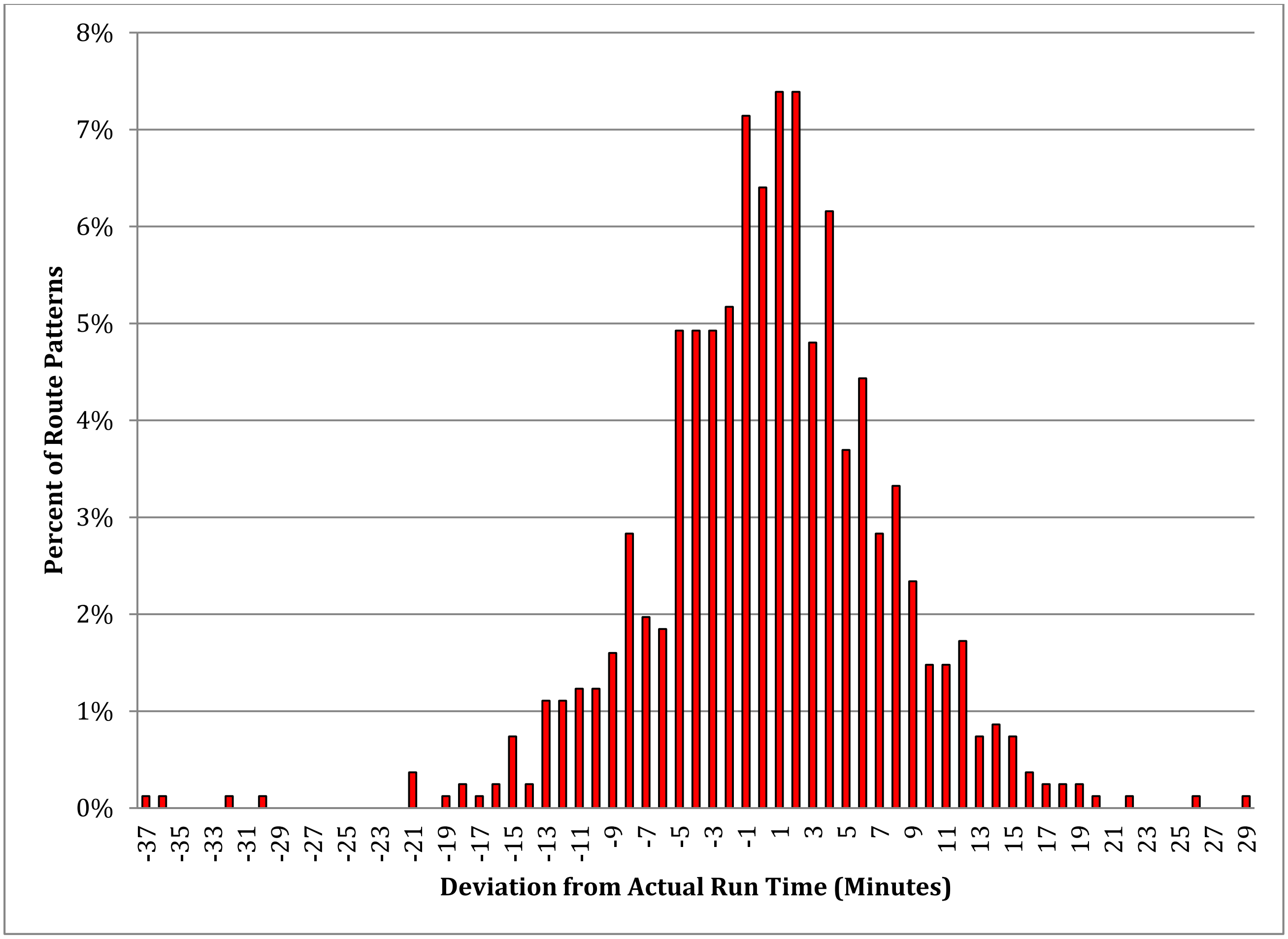
	Area Type						
Assignment Group	**1**	**2**	**3**	**4**	**5**	**6**	**10**
1	56	56	55	54	55	55	56
2	58	58	55	54	55	55	58
3	27	27	26	25	24	26	27
4	19	27	27	26	25	28	19
5	16.5	19.5	18.5	13.5	11.5	17.5	16.5
6	17.5	17.5	17.5	13.5	9.5	16.5	17.5
7	14.5	14.5	13.5	11.5	9.5	13.5	14.5
8	58	58	55	54	55	55	58
10	50	50	50	50	50	50	50
11	45	45	45	45	45	45	45
13	37	37	36	35	35	39	37
14	37	37	36	35	35	39	37
15	47	39	32	28	23	32	47

<u>Route/Pattern Transit Run Time Review</u>

The modeled transit times for each route/pattern were compared to the route times calculated from the patterns. The modeled times are based on the area type/facility type speed lookup function previously described, with the exception of the link-specific times for the fixed rail alternatives (Routes 55 and 888) which are based on stop-to-stop schedule or other data. Figure 9 depicts the fit of the modeled route patterns to the observed travel times. The numeric average modeled pattern is -1.6 minutes slower than the observed, with 33 percent of patterns within +/- 2.5 minutes and 60 percent of patterns are within +/- 5.0 minutes of actual time. On a percentage basis, 60 percent of patterns are within +/- 10 percent of their actual time. The fit of the model is limited by the current use of global speeds and generalized relationships with the highway network (which similarly uses global area type /facility type speed lookup functions rather than actual speeds). Some difference may also be due to simplification of patterns within the modeled network.

Because these speeds are based on tabled lookup functions, deviation from individual route travel time may occur and should be reviewed at the project level (similar to the review that takes place for highway projects).

Figure 9 – Difference between Modeled and Actual Speeds by Pattern



Walk Accessibility

Walk access data was developed for the new TAZ transit system (1201 zones) within the Seven County Metropolitan Area. The percent of trips were calculated that originate in the following three buffer classes: below 1/3 mile, between 1/3 and 1 mile, and above 1 mile of the transit station. The methodology included determining daily trip generation factors by acreage for all existing land use types, and calculating trip generation capacity of the TAZs within the specified buffer zone classes. Table 6 shows the estimated daily trip generation rates, which were determined by using the ITE Trip Generation Manual and typical floor-area ratios.

Table 6 – Estimated Daily Trip Generation Rates

Land Use	Daily Trip Rate (trips/acre)	Land Use	Daily Trip Rate (trips/acre)
Agricultural	0	Mixed Use Commercial	327
Farmstead	0	Industrial or Utility	65
Seasonal/Vacation	0	Extractive	0
Single Family Detached	26	Institutional	261
Single Family Attached	26	Park, Recreational, Preserve	0
Multifamily	78	Golf Course	0
Manufactured Housing Parks	78	Major Highway	0
Retail and Commercial	327	Railway	0
Office	261	Airport	18
Mixed Use Residential	78	Undeveloped	0
Mixed Use Industrial	65	Water	0

USERS GUIDE

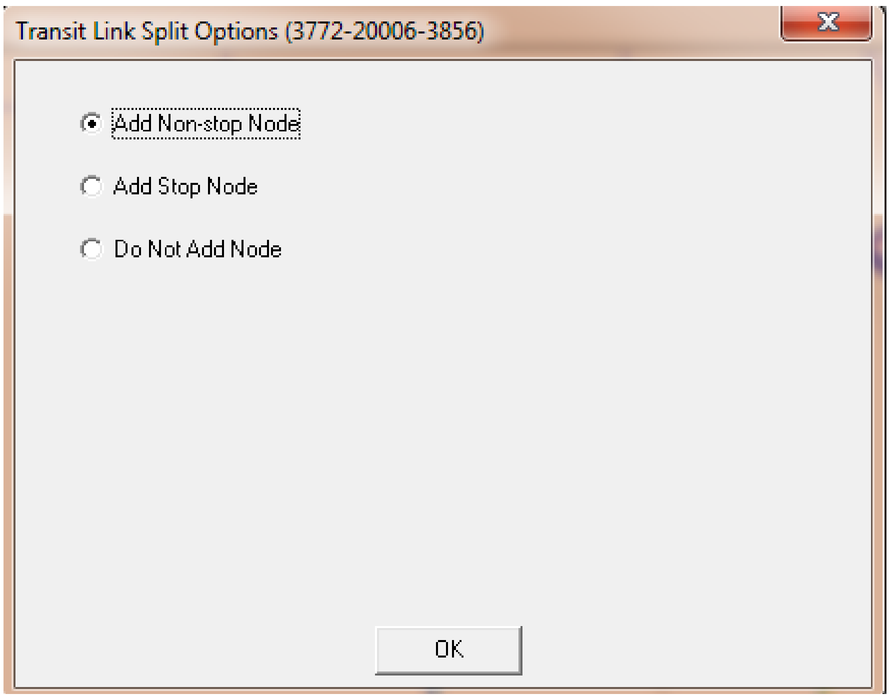
As part of the transit model update and conversion to PT, the process for editing both the highway and transit network have changed. PT requires that each transit A-B link have a corresponding highway network link. The following summarizes the procedure and considerations for editing both the highway and transit networks. **With this update, the regional highway and transit feature classes are synchronized and must be edited concurrently to maintain consistency.**

Highway Network Editing

The updated 2010 PT master transit network is synchronized with the master highway network. When editing the highway network, start by opening the "TransitLines_Master" master transit network feature class. This will open both the master transit network and the corresponding highway network "HighwayNetwork_Master." The transit network can be hidden by checking the boxes in the table of contents. Network editing can be conducted with the network editor procedure described in the Cube help menu.

The basic network editing commands of link split, link dissolve/node delete and node move each affect the transit network are updated in the transit network differently. When *links are split*, CUBE will provide a prompt asking the user to select how the new node should be added to the transit network (Figure 10). A stop or non-stop node should always be selected. If the "Do Not Add Node" command is selected, the transit skims may crash when the model is run. This interface will help maintain geometric consistency between the transit and highway network. When *links are dissolved*, or nodes are removed, the transit network will need to be updated in one of two ways. The node can be removed from the transit feature class or if a large number of nodes are to be removed, find and replace can be used on the transit text line file. This requires exporting and importing the transit feature class. Due to the manual nature of this process, link dissolves should be avoided where possible. When *nodes are moved*, the transit lines will update automatically and no additional action is required.

Figure 10 – Highway Editor Transit Stop/Non-Stop Prompt



Transit Network Editing

Similar to highway network editing, transit network editing should be performed maintaining consistency with the master highway network. When editing the transit network, start by opening the "TransitLines_Master" master transit network feature class. This will open both the master transit network and the corresponding highway network "HighwayNetwork_Master." Transit line editing should be completed with the CUBE GIS transit editing tools. See the CUBE help menu for additional details regarding transit network editing.

The transit line file, "TransitLines_Master," represents a master transit network file. Within the model process, the master transit network is separated into scenario specific networks based on the table "TransitLine_All_Scenario_Years", stored in the geodatabase. This process requires a unique line for each line that changes between the existing and future scenarios. Changes to stop locations, headway and mode require a unique transit line for each modeled scenario. As the transit master network is edited, the "TransitLine_All_Scenario_Years" table must be updated for each of the scenarios to be modeled. Each column in this table provides a complete list of all transit lines to be used for the specified scenario. The following figure provides an example of the table format.

Figure 10 – Example of "TransitLine_All_Scenario_Years" Table

OBJECTID	LINES_2010	LINES_2015	LINES_2020	LINES_2025	LINES_2030
782	856_SB_5GAR	856_SB_5GAR	856_SB_5GAR	856_SB_5GAR	I35WBRT_SP_F
783	860_SB_CRNS	860_SB_CRNS	860_SB_CRNS	860_SB_CRNS	LINES_2030
784	87_NB_ROSE	87_NB_ROSE	87_NB_ROSE	87_NB_ROSE	SnellingNB_F
785	87_SB_KEFO	87_SB_KEFO	87_SB_KEFO	87_SB_KEFO	SnellingSB_F
786	887_EB_SCTC	887_EB_SCTC	887_EB_SCTC	887_EB_SCTC	SW_Exp_F
787	887_WB_BLST	887_WB_BLST	887_WB_BLST	887_WB_BLST	SW_LRT_E_F
788	888_NB_BAPK	888_NB_BAPK	888_NB_BAPK	888_NB_BAPK	SW_LRT_W_F
789	888_SB_BAPK	888_SB_BAPK	888_SB_BAPK	888_SB_BAPK	WBrdwy_NB_F
790	889	889	889	889	WBrdwy_SB_F
791	9_EB_A_1	9_EB_A_1	9_EB_A_1	9_EB_A_1	
792	9_EB_A_2	9_EB_A_2	9_EB_A_2	9_EB_A_2	
793	9_EB_A_3	9_EB_A_3	9_EB_A_3	9_EB_A_3	
794	9_EB_A_4	9_EB_A_4	9_EB_A_4	9_EB_A_4	
795	9_EB_A_5	9_EB_A_5	9_EB_A_5	9_EB_A_5	
796	9_EB_A_6	9_EB_A_6	9_EB_A_6	9_EB_A_6	
797	9_EB_PLGA_1	9_EB_PLGA_1	9_EB_PLGA_1	9_EB_PLGA_1	
798	9_EB_PLGA_2	9_EB_PLGA_2	9_EB_PLGA_2	9_EB_PLGA_2	
799	9_WB_B	9_WB_B	9_WB_B	9_WB_B	
800	9_WB_C	9_WB_C	9_WB_C	9_WB_C	

In addition to the master transit feature class, two additional feature classes have been provided.

- TransitLines_2010_MOD
- TransitLines_2030_MOD

These two feature classes are not required to run the transit model, but may provide the user with additional flexibility in transit network coding. Transit lines added to these feature classes are combined with the corresponding scenario specific transit lines from the master transit network during the model run. Transit lines with a name repeated from the transit master network will replace the transit master network lines. The *TransitLines_2010_MOD* feature class is used in the 2010 scenario and *TransitLines_2030_MOD* feature class is used in the 2030 scenario only.

If, during the course of highway or transit network editing, spatial consistency between the highway and transit layer is lost the spatial display can be restored by exporting and then re-importing the transit master network.

The 2010/2030 PT transit network includes several additional fields as described in Appendix A.

Transit Network Skimming and Assignment

Citilabs currently provides two transit modeling programs, TRNBUILD and PT. A white paper comparison of TRNBUILD and PT was prepared by Citilabs in order to understand the differences in results between TRNBUILD and PT (Attached Memorandum). The 2003 Twin Cities transit model utilized TRNBUILD. The 2010 Twin Cities Transit network and model catalog has been developed in PT. This PT catalog, parameter files and inputs and output files are described in the attached memorandum. The PT model and network were developed to be convertible to TRNBUILD.

Considerations for Future Network Updates

Several characteristics of the available GIS or other information increased the time and resources required for the network update. An effort should be made to correct the following in future network or geodatabase updates to make it more efficient to maintain the integrated transit/highway network:

1. Metro Transit provided a pattern shapefile consistent with trip schedule data but some additional work was required to generate a unique route pattern and correct several spatial discontinuities.

2. Metro Transit pattern shape file could be improved such that pattern spatial direction is consistent with the direction of the routes.

3. Centerline inconsistencies between highway and transit network inputs are perhaps unavoidable. The transit patterns in the Metro Transit Pattern shape file are represented on the actual centerline for divided arterials while the highway network is typically an approximation.

4. The transit modes for this update were defined consistent with the current FORTRAN mode choice program. Future model updates should consider creating separate modes for arterial BRT and highway BRT since they have different speed profiles compared to other transit lines on the same street and different modal bias.

Appendix A

Table A – Highway and Transit Network Attribute Summary

	Attribute	Description	Value	Value Description
Network Link Attributes	T_PRIORITY	Identifies Roadway Segments with Special Transit Characteristics	1	Transit Only Link not used for Highway Model
			2	Roadway Segment Includes Bus Only Shoulders
			3	HOV/HOT Lanes
			4	Roadway Segment Includes Bus Only Lane
	T_MANTIME	Manually Defined Transit Time (Minutes)		
	A_OLD	Metropolitan Council Network A Node Prior to Transit link Segmentation		
	B_OLD	Metropolitan Council Network B Node Prior to Transit link Segmentation		
Transit Network Attributes	NAME	Transit Line Name (28 character limit)		Example: 783_EB_BALW
	LONGNAME	Pattern Identifier that Relates to Metro Transit Schedule Database		Example: BALWMA1050
	UserA5	Route Number		Example: 783
	MODE	Identifies Transit Line Type, Fare, Bias	5	Local Bus
			6	Suburban Local Bus
			7	Express Bus
			8	Light Rail
			9	Commuter Rail
	Operator	Owner of Transit Line	1	Prior Lake, Shakopee, or BlueXpress Joint Powers
			2	Maple Grove Transit
			3	Metro Transit
			4	Minnesota Valley Transit Authority
			5	Plymouth Metrolink
			6	Ramsey Star Express
			8	Southwest Transit
			9	University of Minnesota
	ONEWAY	Identifies Pattern as one or two way	1 or True	All transit lines and drive access non-transit links are one-way
			2 or False	All walk access non-transit links are two way
	Headway_1	AM Headway (Minutes)	1-180	See Table 3 for list of Headway Values
	Headway_2	Midday Headway (Minutes)	1-360	See Table 3 for list of Headway Values

Appendix B

Metropolitan Council Regional Model Transit Network Update: TRNBUILD to PT Conversion

Contents

About this Document

This document presents a “white paper” prepared by Citilabs on various topics of relevance to the conversion of transit networks from the legacy TP+ TRNBUILD format to the newer Cube Voyager Public Transport (PT) format. This white paper was specially written for the Metropolitan Council by Citilabs under a subcontract to SRF Consulting.

Topics include:

- TRNBUILD and PT formats and coding standards
- Transit path-building, skimming, and assignment in PT versus TRNBUILD
- Feasibility of crowd modeling and schedule-based transit analysis in the Twin Cities

Introduction to Public Transport

Public Transport (PT) is the Cube Voyager program module for transit analysis, providing a comprehensive system for multi-modal network development, skimming, and assignment. PT is designed to replace the transit analysis routines in legacy Citilabs software such as TRIPS, TRANPLAN, and TP+, including the TRNBUILD program which is currently used in the Twin Cities Regional Travel Demand Model. As such, PT borrows concepts and features implemented in or developed for such legacy packages. For example:

- At the core of PT is an innovative logit-based multi-path transit route enumeration and evaluation process originally developed for TRIPS 2000.
- PT includes support for detailed stop-to-stop reporting capabilities, a popular TRANPLAN feature that was not available in TRNBUILD.
- The Cube Base and Cube Voyager user interfaces to PT extend conventions established in Viper/TP+, with a similar line and access/egress/transfer link coding format, as well as a familiar command/keyword scripting syntax.

Accordingly, while some aspects of PT will be familiar to experienced users of any one of these three legacy software packages, important differences do exist which might influence the upgrade path of agencies like the Twin Cities Metropolitan Council. The purpose of this document is to identify such differences and describe an upgrade path designed to allow the Metropolitan Council to move to PT and take advantage of the powerful new features it offers at minimum cost and difficulty.

Public Transport Network Coding

The central data structure manipulated by PT is a multimodal transportation network. This includes the following components, which work together to form an integrated whole:

- A Cube Voyager/TP+ **highway network** consisting of a table of nodes (points) and links between these, optionally enhanced to exclude prohibited links and include links for non-auto infrastructure, such as sidewalk links or transit-only guideways.
- A set of **public transport lines** representing the services (or routes) operated by transit providers within a model study area. Each line consists of a connected sequence of nodes associated with line-level, link-level, and node-level attributes.
- A set of **non-transit legs** to represent access, transfer, and egress connections between zone centroids and stop nodes. These may be generated by PT using network path-building logic, or they may be hand-coded by the user in Cube Base.

Other data processed by PT include the following:

- **Fare systems** are described in a text-format file using Cube Voyager script command/keyword syntax, which may make reference to network and line attributes and their values.
- **Factors** influencing the route enumeration and evaluation processes are coded in a separate input text file for each user class to be analyzed, also using the Cube Voyager command/keyword script syntax.

In Cube 5.0 and later, the highway network, public transport lines, and non-transit legs can be stored in ESRI ArcGIS formats as well as legacy binary and text formats. Currently, all ESRI geodatabase types are supported by Cube, including personal (MDB), file (GDB), and multi-user (SDE) geodatabases. All of these employ similar relational GIS concepts to associate nodes, links, and lines based upon attribute information. The characteristics of the different types of geodatabases are compared and summarized in the table below.

Characteristics of ESRI Geodatabase Types

Geodatabase Type	Database Technology	Editing Quality	Limitations
Personal (MDB)	Microsoft Access	No versioning; slow	Total 2 GB storage limit

File (GDB)	None (folder structure)	No versioning; faster	1 TB per table size limit
Multi-user (SDE)	Microsoft SQL Server Express or Enterprise Relational Database Management System	Supports versioning, replication, & archival; speed depends on underlying database	Requires ArcInfo, ArcEditor, or ArcGIS for Server license; storage capacity depends on underlying database

While the ESRI geodatabase formats supported by Cube offer direct compatibility with ArcGIS software and many other benefits including superior mapping and geoprocessing capabilities, as well as the ability to synchronize master highway network edits with transit lines, the PT binary and text file formats are generally faster and easier to work with for pure transportation modeling applications. Furthermore, they are directly comparable with the TP+ file formats currently used in the Twin Cities Regional Model, whereas the TRNBUILD program does not support ESRI geodatabase storage. There are currently no plans to add geodatabase support to TRNBUILD or any other legacy TP+ program.

The PT line text file format consists of a series of statements in Cube Voyager script syntax containing one or more LINE commands followed by keywords which either specify attributes of the line or provide a list of connected nodes representing the route followed by the transit service being described, with optional sub-keywords available to specify node-level and link-level service attributes. In other words, PT uses LINE statements in much the same manner as TP+, with the following exceptions:

- Line files are typically incorporated into the body of TRNBUILD scripts using the READ FILE statement, whereas PT designates a separate FILEI LINEI command. This means that non-LINE commands and keywords are not valid within PT line files; for example LINK statements may not be incorporated.
- In TRNBUILD the line service frequency was coded using a FREQ or FREQUENCY keyword; however the value actually coded was defined as the time between successive trips or runs of the coded route, which is actually the *headway* (i.e. the inverse of frequency). PT corrects this discrepancy by re-naming the FREQUENCY keyword to HEADWAY (with identical function).

Since PT is newer and being continually updated by Citilabs on an ongoing basis as a major component of the Cube Voyager system, the PT LINE file format also offers several optional features that were not supported in TRNBUILD:

- The LINE NODES NNTIME sub-keyword allows the user to encode node-to-node travel times within the line definition, overriding highway network assumptions.

- The CIRCULAR keyword can be used to code lines that have no defined starting and ending point, such as downtown circulators. In other words, the program recognizes that it is possible for a valid trip on such a line to have an alighting stop which occurs before the boarding stop within the line's node list, as coded.
- The OPERATOR keyword allows the user to identify different transit service providers for different routes (such as Metro Transit and MVTA).
- The FARESYSTEM keyword allows the user to differentiate fares by line as well as by mode or by operator (TRNBUILD only allowed fares assessed by mode or link).
- PT includes a crowd modeling algorithm that permits the user to perform transit assignment with capacity restraint (only available when using multi-path analysis). Keywords associated with this capability, not previously available in TRNBUILD, include VEHICLETYPE, SEATCAP, CRUSHCAP, and CROWDCURVE.
- PT also now offers an alternative mode of analysis in which detailed timetable data rather than general headways by time period are used to characterize the level of service offered on each line, and to include the assessment of timed transfers between infrequent and connecting services. Keywords associated with this capability, not previously available, include STARTTIMES and INTERVAL.

The non-transit leg text file format is also very similar to the support link text file format used in TRNBUILD. Superficially, a support link file can be converted into a non-transit leg file by searching and replacing all occurrences of "SUPPORT LINK" with "NT LEG". However, there are some fundamental differences between TRNBUILD and PT in the assumptions behind the generation and use of access, transfer, and egress links. All zone-to-zone transit paths developed by PT must in theory consist of alternating non-transit and transit legs, meaning that a path containing two sequential non-transit legs should never be constructed by the program. The transit networks previously used in the regional model violated this principle; however, after this issue was raised early on by Citilabs staff, SRF Consulting addressed the problem in their re-coding of the 2010 transit network.

Interim Twin Cities Transit Network Coding Standards

Because Cube Voyager can be used to read transit network data from an ESRI geodatabase and write the same information to a file in one of the legacy binary or text file formats, it is possible to use the geodatabase for data storage and maintenance while continuing to use proven TRNBUILD programs for transit network skimming and assignment. This approach is suggested as an interim solution for backwards compatibility while developing and testing new PT-based skimming and assignment procedures. In order to apply this approach, however, the technician coding the transit networks must adhere strictly to certain rules based upon the feature comparison described above, in order to ensure that the resulting data structures are “platform-independent”. If these coding standards are followed, then the line and access/transfer/egress data stored in PT format within an ESRI geodatabase can easily be extracted to text files and then automatically converted to TRNBUILD format using global search-and-replace of “HEADWAY” with “FREQ” and “NT LEG” with “SUPPORT LINK”.

The following table summarizes the key areas of non-overlap between TRNBUILD and PT, presenting these as “prohibited practices”, grouped by the platform on which they would be permitted (i.e. either TRNBUILD or PT).

Platform-Specific Coding Practices Avoided in New Twin Cities Transit Networks

TRNBUILD	• Non-LINE commands and keywords in LINE data • Sequential (non-alternating) non-transit legs.
PUBLIC TRANSPORT (PT)	• LINE NODES NNTIME • CIRCULAR lines • OPERATOR and FARESYSTEM • Crowd modeling using VEHICLETYPE, SEATCAP, CRUSHCAP and CROWDCURVE • Time-tabling using INTERVAL and STARTTIMES

These standards were communicated to SRF Consulting staff by Citilabs at the outset of this project and used to code new geodatabase-driven 2010 transit networks. Thus, it is currently feasible to use the PT format for storage of Twin Cities regional model transit data. The next section addresses whether it is feasible to use the PT programs to replicate the transit skimming and assignment steps currently being performed by TRNBUILD in the regional model.

Routing, Skimming and Assignment

As mentioned in the introduction to this document, PT incorporates many innovative concepts in transit path-building that were developed previously for the TRIPS 2000 legacy software update. As such, the underlying framework for PT routing is fundamentally different from that used by TRNBUILD, and specifically, originates from a modelling tradition that developed within the context of the more matured and varied portfolio of public transport services operated in European cities, with complex policy questions to be addressed compared to many North American transit systems.

Recently, the emergence of global markets for commercial transportation planning software and travel demand modeling community has contributed to significant advancements that have been made and are ongoing in American travel demand analysis. To respond to these emerging, complex needs, to maintain a level of basic functionality , as well as provide an agency and market-driven product Citilabs remains in communication with planning and regulatory agencies to apprise them of ongoing developments in Cube and to respond to their needs. For example, in response to feedback from the Federal Transit Administration, Citilabs implemented the BESTPATHONLY parameter, which restricts the set of paths evaluated to the single best option among all those enumerated. In general, this all-or-nothing skimming and assignment approach is similar to the default behavior of TRANPLAN or TRNBUILD, which is to build, analyze and assign to one and only one path for each origin-destination pair that is connected by the mode of service defined by the user. Accordingly, Citilabs recommends that modelers migrating from TRANPLAN or TRNBUILD to PT use the BESTPATHONLY parameter when comparing skimming or assignment results with previous versions of their model.

Skim Functions and Their Equivalents

All of the TRNBUILD functions currently used to develop skims for mode choice in the Twin Cities Regional Model have directly equivalent functions in PT. These are summarized in the table below.

TRNBUILD-PT Comparison for Local and Limited-Stop Bus Routing

Skim definition	TRNBUILD skimming function	Equivalent PT skimming function
IVTT for local bus (mode=5) & local limited stop bus (mode=6)	TIME(5, 6)	TIMEA(0, 5, 6)
Initial wait time	IWAIT	IWAITA(0)
Transfer wait time	XWAIT(5, 6)	XWAITA(0)
Number of boardings	BOARDS	BRDINGS(0, 5, 6)
OVTT for walk access & transfer	TIME(1, 4)	TIMEA(0, 1, 4)
Fare on express (mode=7), LRT (mode=8), or CRT (mode=9)	MAXMODEFARE	MAX(FAREA(0,5),FAREA(0,6),FAREA(0,7))

Some additional TRNBUILD skim functions that are not directly used in mode choice do not have equivalent functions in PT. These include MODET1, NODE0, and NODEL. However, these functions can and have been emulated in Cube Voyager by writing and post-processing a PT stop-to-stop report table using the MATRIX script shown on the following page.

```
RUN PGM=MATRIX MSG='Skimming for Station Data'
FILEI MATI[1] = "{CATALOG_DIR}\Model\AM_LOCAL_WK_SKIM_TMP1.MAT"
FILEI DBI[1] = "{CATALOG_DIR}\Model\AM_LOCAL_WK_S2S_1.DBF",
 SORT=I,J
FILEI DBI[2] = "{CATALOG_DIR}\Model\AM_LOCAL_WK_S2S_2.DBF",
 SORT=I,J,FROMNODE
FILEI DBI[3] = "{CATALOG_DIR}\Model\AM_LOCAL_WK_S2S_2.DBF",
 SORT=I,J,MODE
FILEO MATO[1] = "{SCENARIO_DIR}\2030AM_LOCAL_WK_SKIM_PT.MAT",
 MO=1-13 DEC=13*4
NAME=IVT,WAIT1,WAIT2,XFERS,WALKT,FARE,FIRST_MODE,1ST_MODE5,1ST_MODE6,
 1ST_MODE7,LAST_MODE5,LAST_MODE6,LAST_MODE7

ARRAY FROMSTOP={TOT_ZONES},{TOT_ZONES},10 TOSTOP={TOT_ZONES},{TOT_ZONES},10

FILLMW MW[1]=MI.1.1(6)            ; IVT,WAIT1,WAIT2,XFERS,WALKT,FARE

IF (I=1)
   LOOP _I=1,DBI.2.NUMRECORDS
      TMP=DBIREADRECORD(2,_I)
      FROMSTOP[DI.2.I][DI.2.J][DI.2.MODE]=DI.2.FROMNODE
      TOSTOP[DI.2.I][DI.2.J][DI.2.MODE]=DI.2.TONODE
   ENDLOOP
ENDIF

JLOOP
   _SEEK1=DBISEEK(1,I,J)           ; for I & J zones
   IF (_SEEK1=0)
      _FROMNODE=DI.1.FROMNODE      ; first station node number
   ENDIF
   _SEEK2=DBISEEK(2,I,J,_FROMNODE) ; for I zone, J zone, first station node
   IF (_SEEK2=0)
      MW[7]=DI.2.MODE              ; mode number for first station
   ENDIF

   IF (FROMSTOP[I][J][5]>0)
      MW[8]=FROMSTOP[I][J][5]      ; first station for mode 5
      MW[11]=TOSTOP[I][J][5]       ; last  station for mode 5
   ELSEIF (FROMSTOP[I][J][6]>0)
      MW[9]=FROMSTOP[I][J][6]      ; first station for mode 6
      MW[12]=TOSTOP[I][J][6]       ; last  station for mode 6
   ELSEIF (FROMSTOP[I][J][7]>0)
      MW[10]=FROMSTOP[I][J][7]     ; first station for mode 7
      MW[13]=TOSTOP[I][J][7]       ; last  station for mode 7
   ENDIF
ENDJLOOP

ENDRUN
```

MATRIX Post-Processing Script to Emulate MODET1, NODE0, and NODEL Skims

Comparability of Skim Results

Even with the BESTPATHONLY parameter turned on and the equivalent skim functions identified, there are some significant underlying differences in the ways that TRNBUILD and PT build and evaluate paths which translate into apparent differences in skim results. Specifically, when there are multiple lines available in a route bundle (e.g. serving the same boarding-alighting stop pair), TRNBUILD and PT may choose different lines in a path segment, even though the paths built are the same in terms of stop-to-stop movements. In PT, the basic set of lines considered is built using the Service-Frequency model described below.

By contrast, in TRNBUILD the basic set is built using the LINE COMBINATION procedure as outlined below:

1. Select best line: line with B(Ptt)
2. Select reasonable lines: check COMBINE criteria,
   a. Ptt - B(Ptt) <= MAXDIFF
   b. WHEN Ptt - B(Ptt) >MAXDIFF, check IF condition
3. **Path wait time**: half of headway, i.e., WaitFactor* 60/sum(selected lines' frequency)
4. **Path run time**:
   a. Revised perceived wait time for each line:

      RPWait = Ptt - B(Prun)

   b. Compute weight factors:

      For Line x: Weight of line x = revised frequency of line x/ sum(revised frequency of all selected lines)

   c. Perceived Path run time:

      PPrun = Sum(Weight*Prun)

where:

Actual Wait Time = Await; Perceived Wait Time = Pwait = Await*WaitFAc

Actual Run Time = Arun; Perceived Run Time = Prun = Arun*ModeFac;

Perceived travel time = Ptt = Prun + Pwait;

Best Value = B(argument), e.g. B(Prun), B(Ptt)

It can easily be seen by comparing this logic with that described in the PT documentation that they are obviously different. Therefore, even though the shortest paths are the same, the basic line sets, wait times, and skim times resulting from these paths may be quite different. In some cases, for example, the differences in wait times could result in a path being found by PT that would not be considered acceptable by TRNBUILD because the wait time (as calculated differently) would be in excessive of the maximum threshold. Even on paths where both programs do find a path, the skim values could be different, due to these calculation differences.

In an attempt to assess the upper bound on the magnitude of the potential impact of these skim differences on the regional model, Citilabs performed a test by multiplying the total work purpose person-trip table from a recent 2010 model run (provided by SRF Consulting) times the total travel time difference between the transit paths skimmed by TRNBUILD and PT, to gauge the potential magnitude of the impact of these differences on transit ridership forecasted by the Twin Cities model. According to this test the maximum (upper bound) increase in person-hours of travel by transit reported by PT versus TRNBUILD is roughly 1.4 percent. The total number of person-trips on origin-destination movements where PT constructed a path and TRNBUILD did not or vice-versa is 10,379.51, or 0.6 percent of the total work purpose person-trip matrix. By either criterion the difference is apparently “small”, although once it would be necessary to run the full regional model with all TRNBUILD steps replaced by PT steps in order to truly evaluate the impact.

There is also reason to believe that the paths constructed and skimmed by PT may be more accurate than those provided by TRNBUILD. Using a trip table compiled from transit on-board survey data provided by SRF Consulting, Citilabs was able to identify 652 person-trips on origin-destination pairs for which PT provided a valid path, but TRNBUILD did not. This suggests that in fact the path-building process programmed in the legacy software may not be in accordance with the empirical evidence of transit travel for some origin-destination pairs, whereas PT may provide a more accurate evaluation of which paths should be considered valid, compared to observed trip patterns. In this case, it may not be advisable to reverse engineer or otherwise force PT outputs to match previously obtained TRNBUILD results. Furthermore, any differences in results might be outweighed by the greater functionality available in PT, including new features such as crowd modeling and timetabling, which are described in the next section.

Service-frequency model

Applied at stops with a basic set of transit choices, the service-frequency model calculates the conditional probabilities of the individual lines in proportion to their frequency. The model is equivalent to travelers who arrive randomly without knowledge of the timetables and take the first reasonable service forward from the node.

$$P_{(walktoi)} = \frac{e^{-\lambda(CW_i + \alpha ECD_{ai})}}{\sum_j e^{-\lambda(CW_i + \alpha ECD_j)}}$$

where:

use line l)	is the probability of using line l.
Frequency(line l)	is the frequency of line l (in services per hour).
Frequency(line k)	is the frequency of line k (in services per hour).

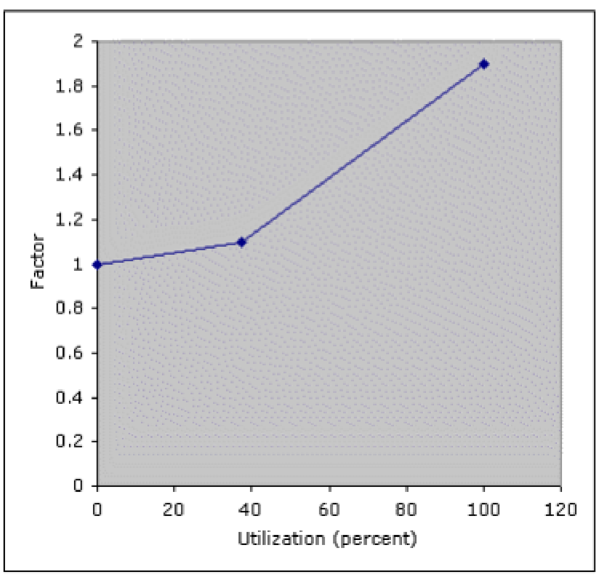
Advanced Modeling Features

In addition to the more traditional modeling applications of PT using best-path analysis, several extensions of its multi-path analysis functionality suggest the possibility of new uses for the Twin Cities transit network. In particular, the crowd modeling and time-tabling features of PT offer quasi-“operational” modes of analysis that may be of special interest to the Metro Council, since it also provides service through its Metro Transit and Metro Mobility divisions. This section assesses the feasibility of implementing crowd modeling and timetabling analysis in the Twin Cities for selected applications.

Crowd Modeling

In most urban areas in the United States, crowding phenomena are generally seldom observed on most transit service, due to low ridership. However, ridership has increased in recent years with increasing gas prices, as well as with increasing availability of rail and bus rapid transit options. Furthermore, while crowding might not occur during an average or typical travel day, crowding can be a defining characteristic of special event travel, such as at sports or entertainment venues where parking costs can spike in response to demand.

Crowding is analyzed in PT by associating a vehicle type with each line, and defining the seated and crush load capacity for each vehicle type. The user furthermore specifies a “crowding curve” which represents the factor by which perceived travel increases due to the inconvenience of standing (see below for example).



Crowding Curve Definition, as a Function of Utilization

The definition of the utilization measure used as input to this curve is:

$$U = \frac{P - (LDF_v\ SeatCap)}{CrushCap - (LDF_v\ SeatCap)}$$

where:

P is the passenger demand (per hour).

$SeatCap$ is the seating capacity (per hour) for the line.

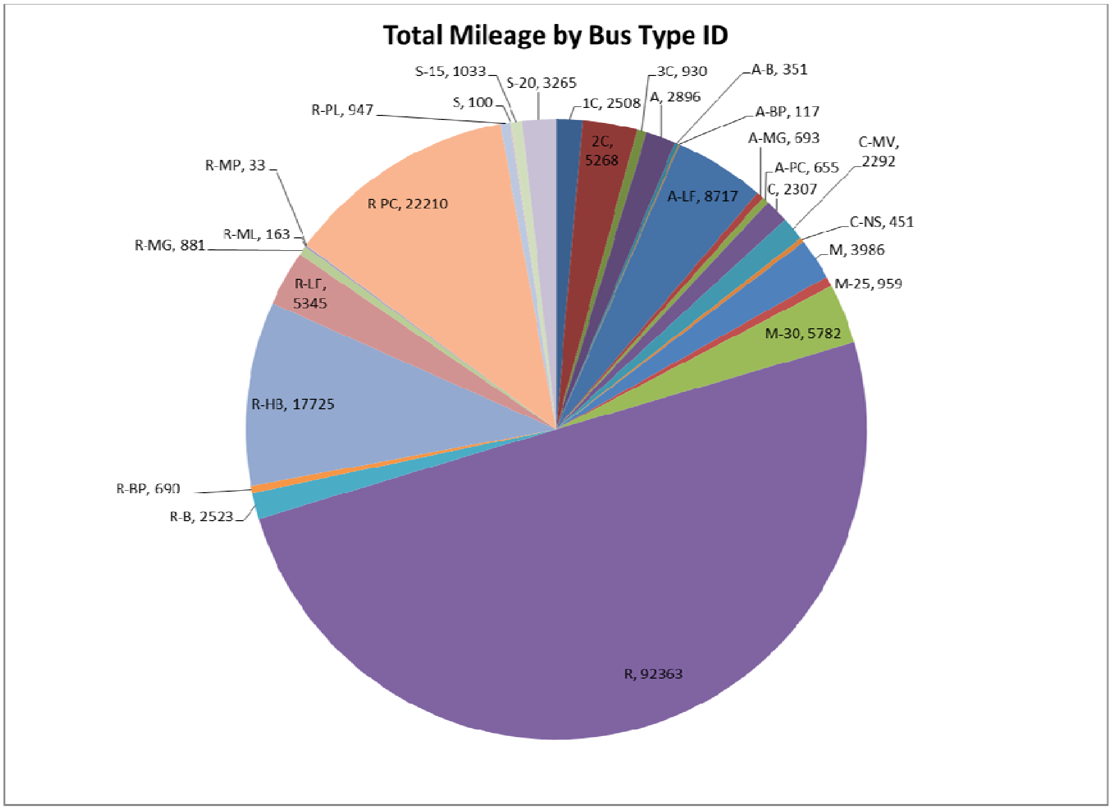
$CrushCap$ is the crush capacity (per hour) for the line; this is the total of seated and standing capacities.

LDF_v is the load distribution factor for the vehicle type. As loads increase this is the proportion of seats occupied when standing starts to occur. It may be less than 1.0, indicating that standing occurs before all seats are used. (Note: the LDF value specified in commands is coded as the corresponding percentage.)

The value of utilization is constrained with a minimum value of 0% (when no standing occurs) and an upper limit of 100% (when crush capacity is exceeded). However, even after the crush capacity is exceeded, perceived travel times may increase with demand due to adjustments to wait time as passengers are forced to board the next appropriate service that has available capacity, rather than choosing any appropriate service. This adjustment can divert riders from the shortest transit path to other lines or routes; for this reason, the crowd modeling feature is only available with multi-path analysis. With crowd modeling, routes are re-evaluated iteratively, to perform a capacity-restrained transit loading similar. If no alternative paths are available, then a crowded transit leg can become a bottleneck and not all of the input demand may be assigned during the modeled period. The PT crowding model recognizes the additional delays imposed on travelers forced to wait so long that they are not able to make their trip during the model period, as well as the “flow metering” effects of demand being lower downstream due to travelers being unable to pass the bottleneck location. The EXCESSDEMAND skim function can be used to report the number of trips for each origin-destination pair that could not be loaded due to capacity bottlenecks in the transit system. This function may be especially useful when planning service for special events or evacuation conditions.

The data requirements in order to perform a PT crowding analysis include information regarding the vehicles operating on each line, and their seated and crush load capacities. The new 2010 transit networks prepared by SRF Consulting are linked using a “pattern” field to the Metro Transit HASTUS trip database that does include a “bus_type_id” field, which is populated for every line (including the 55

Hiawatha Light Rail service). The overall distribution of service mileage by bus type is shown in the graphic on the following page.



Thus, it does indeed appear to be feasible to define a vehicle type for each line in the new GIS-driven 2010 Twin Cities transit network, and it may even be possible to populate the VEHICLETYPE line attributes automatically by performing queries and joins between the PT geodatabase and the Metro Transit trip database. Two additional data items would then need to be supplied in order to perform crowd modeling:

- The seated and crush load capacities of each bus_type, and
- The crowding curve for perceived link travel time adjustment.

Of these two items, the former is much easier to measure than the latter, and may already be readily available from Metro Transit service planners. However, it might be possible to compare reported travel times from the transit on-board survey with actual travel times to assess whether respondents mis-reported their perceived travel time by a consistent factor that could be related to crowding effects on certain lines. In the absence of any local data, a crowding curve could be borrowed from another study in a large metro such as London or Tokyo, or a flat link travel time escalation curve could be assumed, with any delay and diversion solely resulting from adjustments to wait time at bottleneck locations.

Note: Because it requires multi-path analysis, crowd modeling is not currently an accepted methodology in the United States for New Starts project transit ridership forecasting. However, as mentioned previously, it may be of interest for operational applications.

Time Tabling

The current state of the practice in North American transit modeling uses a so-called "headway-based" analysis which characterizes services in terms of the average time between arrivals at any given stop along a line. Traditionally, in legacy software such as TRNBUILD, the wait time at a stop served by a single line is essentially calculated as one-half the headway of that line (the wait time calculation when multiple lines serve a stop can be slightly more complicated using the line combination logic described previously, but still reflects a linear relationship to service frequency). To reflect variations in service frequency throughout the day, multiple time periods may be used. However, this method of calculating wait times becomes increasingly inappropriate as service frequency declines. For example, travelers are very unlikely to actually wait 45 minutes for a commuter rail or express bus service that only makes two runs during a three-hour period. More likely, they would consult publicly available schedule information, and go to their stop to catch the bus or train at some more reasonable amount of time in advance of the service's scheduled arrival there (e.g. 15-20 minutes). Furthermore, in networks with multiple connecting infrequent services, there may be some transfers that would appear to be valid according to a headway-based analysis which simply cannot be made because the downstream service actually leaves before the upstream service arrives.

Historically, such issues have not been of major concern in transit modeling, because the type of service provided overwhelming consisted of generally frequent, mostly local bus service. However, the following trends in the Twin Cities may create a need to consider alternative analysis methods in the future:

- With Northstar commuter rail now established, and additional long-distance rail service to hub stations in downtown Minneapolis and St. Paul being proposed (e.g. Midwest High Speed Rail, Red Rock, Rush Line), a finer-grained analysis may be necessary in order to assess the true attractiveness of these less frequent forms of service, taking into account the quality of connections to other local lines;
- Express commuter busses routed mostly on freeways with demand fed by suburban park-and-ride lots have become more predominant throughout the

metro, and though faster, these services generally offer lower service frequencies (with correspondingly higher headways) than local buses;

- Budget cutbacks and re-structuring to save costs may, in some areas, reduce the frequency of current service, potentially eliminating or delaying certain transfers;

- Metro Transit has explicitly designated a “high-frequency network” of bus and light rail lines that operate with headways of 15 minutes or less, the value of which might be better represented with a more rigorous analysis of infrequent service.

To address these concerns, Citilabs has implemented a schedule based, or “time-tabling” methodology in PT that uses actual schedule data rather than headways to analyze the quality of service in transit networks. This alternative mode of operation enumerates multiple paths departing at various discrete times within the model period and evaluates these in a manner similar to traditional multi-path analysis, resulting in a form of dynamic *transit* assignment (DTA for transit). Since this is a completely different kind of analysis from the traditional headway-based approach, it must be adopted for the entire network; it is not possible to mix headway-based and schedule-based skimming or loading processes within a single PT program run.

To implement schedule-based analysis, the user must first code the exact duration of the model PERIOD (start time and end time, in *hhmm* format) and DAYTYPE to be evaluated (schedules for multiple day types, such as weekday and weekend, may be coded) in the FACTORS file. Then, the user may define schedule information for each LINE in any of several ways:

- The LINE STARTTIMES keyword provides a list or range of departure times (in *hhmm* fomat) of runs from the starting node. If a range is supplied, then a LINE INTERVAL keyword should be specified immediately preceding STARTTIMES to indicate the headway used to impute the list of departure times from the range. This may provide a simple way to translate headway information into schedule assumptions in the absence of better data.

- The NODES RT statement can optionally be used to set exact time points by station which apply to all runs of a given line, to improve the accuracy of estimated wait times and feasibility of timed transfers. Otherwise, travel times will be derived from standard network-based assumptions.

- Alternatively, the PTRUN command can be used within text line files to provide detailed timetable information for each line, with the following keywords:

- OFLINE: specifies which line the run is a part of
- DAYTYPE: specifies what day types the run operates on
- NODES: defines a point where timing data are read as sub-keywords:
    - AT: time(s) when service calls at the stop (in *hhmm* format)
    - ARR: time(s) when service waits at the stop (in *hhmm* format)
    - DEP: time(s) when service departs from the stop (in *hhmm* format)

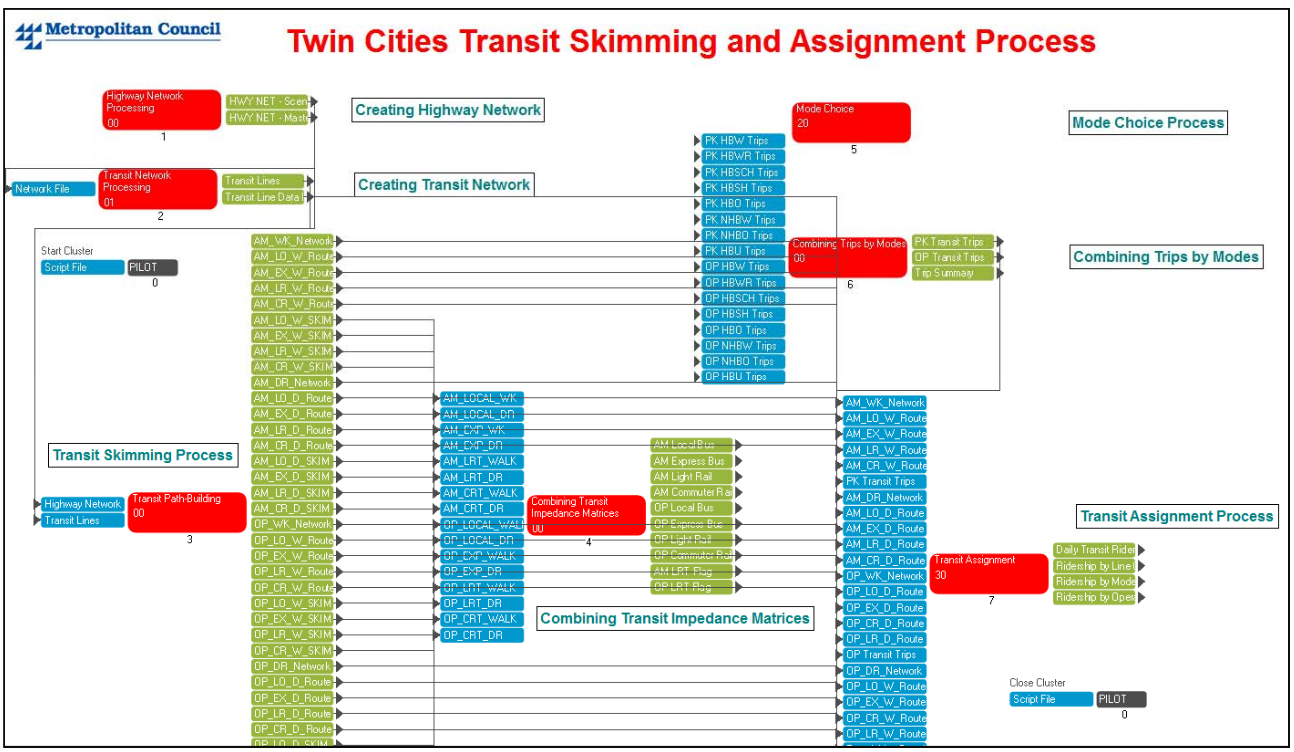
The data requirements for schedule-based transit modeling can be met by the HASTUS trip (pattern) database describing service operations provided by Metro Transit for use in this project. Each run (pattern) of every line is associated with a start time and end time; thus it should be possible, at minimum, to specify an exact list of STARTTIMES for the line, and provide a more accurate RUNTIME for the line as a whole. Since these data were directly used to calculate headways in the new GIS-driven 2010 Twin Cities transit networks prepared by SRF, the process could largely be automated using database queries, joins, and Cube Voyager record processing scripts. Individual time point data for selected stations are also available in a separate table which could potentially be related to the stop nodes in the PT network as well; however this would require some GIS analysis, such as a spatial join.

The data requirements for schedule-based PT analysis could also be met by the general transit feed specification (GTFS) data files used to power the Google Transit trip planning service, which is operational in the Twin Cities. However, the GTFS data were not directly used by SRF Consulting in developing the new GIS-driven 2010 Twin Cities transit networks, primarily because they lacked any spatial component other than simple X and Y coordinates for selected stop locations. The HASTUS databases, by contrast, integrate directly with ArcGIS and hence with Cube Voyager. Furthermore, HASTUS is used internally by Metro Transit for service operations planning purposes as well as their official traveler information services, therefore direct use of these databases would seem to be likeliest to provide the most accurate possible translation of service characteristics into modeling network with the least opportunity for error during translation of formats.

Appendix A Public Transport Skimming and Assignment Application

Citilabs has developed a transit skimming and assignment application for the Twin Cities using the Cube Voyager Public Transport (PT) routine. As part of this effort, options were applied within PT to provide consistency with the TRNBUILD model currently adopted by the Metropolitan Council. The primary goal of this task was to identify settings equivalent to those implemented in the TRNBUILD model, as well as develop additional functions, such as transit master network and ridership summary processing routines. This Appendix serves as documentation for the PT analysis application, to support its potential incorporation in future versions of the Twin Cities Regional Travel Demand Forecast Model (RTDFM).

Figure 1. PT Analysis Application Developed by Citilabs



TRANSIT NETWORK COMPONENTS

1. Transit Modes and Operators

The transit network used for the PT analysis application was developed by SRF Consulting Group. This network includes 3 non-transit and 5 transit mode codes. The first three mode codes represent non-transit legs such as walk-access/egress, drive-access, and transfer. The next five mode codes represent the transit modes, such as local bus, local limited stop bus, etc.

Table 1 shows the list of transit modes, non-transit modes and operators defined in the model. The transit operators are used to summarize the transit ridership.

Table 1. Transit Modes and Operators

MODE NO.	MODE DESIGNATION	DESCRIPTION	OPERATOR NO.	OPERATOR
1	Non-Transit	Walk Access		1= BX 2=MG 3=MT 4=MVTA 5=PLM 6=RS 8=SW 9=UofM
2		Drive Access		
3		Not Used		
4		Stop-to-Stop Transfer		
5	Transit	Local Bus	3, 9	
6		Suburban Local Bus	1, 2, 3, 4, 5, 8	
7		Express Bus	1, 2, 3, 4, 5, 6, 8	
8		Light Rail	3	
9		Commuter Rail	3	

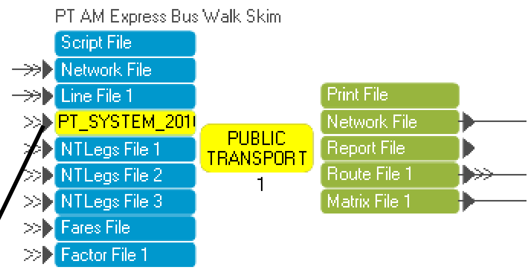
The access-links or non-transit legs (NTL) are generated using hard-coded files provided by SRF Consulting Group. These legs are incorporated during the transit skimming process. Three types of non-transit legs are generated:

1) Walk Access/Egress NTL (mode 1)
2) Drive Access NTL (mode 2)
3) Transfer Access NTL (mode 4)

It should be noted that the non-transit mode 3 used in the former TRNBUILD model to connect parking lots to the transit stops is no longer used in the application developed by Citilabs because PT does not allow origin-destination paths to traverse multiple consecutive non-transit links. Instead, drive-access links in the PT model are directly connected from zones to the appropriate transit stops/stations.

Figure 2 shows a system input file used to describe the characteristics of the public transport system such as modes, operators, and wait curves.

Figure 2. Settings of Transit Modes and Operators




```
;;<<PT>><<SYSTEM>>;;
MODE NUMBER=1 LONGNAME="Walk Access Connectors" NAME="Walk Access"
MODE NUMBER=2 LONGNAME="Drive Access Connectors" NAME="Drive Access"
MODE NUMBER=3 LONGNAME="Transfer Connecotrs" NAME="Transfer"
MODE NUMBER=4 LONGNAME="Stop-To-Stop Connectors" NAME="Stop2Stop"
MODE NUMBER=5 LONGNAME="Local Bus" NAME="Local"
MODE NUMBER=6 LONGNAME="Local Limited Stop Buses" NAME="Local Limited"
MODE NUMBER=7 LONGNAME="Express Bus" NAME="Express"
MODE NUMBER=8 LONGNAME="Light Rail" NAME="LRT"
MODE NUMBER=9 LONGNAME="Commuter Rail" NAME="CRT"

OPERATOR NUMBER=1 LONGNAME="BX" NAME="BX"
OPERATOR NUMBER=2 LONGNAME="MG" NAME="MG"
OPERATOR NUMBER=3 LONGNAME="MT" NAME="MT"
OPERATOR NUMBER=4 LONGNAME="MVTA" NAME="MVTA"
OPERATOR NUMBER=5 LONGNAME="PLM" NAME="PLM"
OPERATOR NUMBER=6 LONGNAME="RS" NAME="RS"
OPERATOR NUMBER=7 LONGNAME="ST" NAME="ST"
OPERATOR NUMBER=8 LONGNAME="SW" NAME="SW"
OPERATOR NUMBER=9 LONGNAME="UofM" NAME="UofM"
```

2. Transit Network Elements

The transit network, as described in the previous section, consists of transit and non-transit legs. The transit legs are built using highway network (or "physical") links, such as rail links for fixed guideway transit. As mentioned previously, the non-transit legs element was built using hard-coded links developed by SRF Consulting Group.

Table 2. Transit Related Files

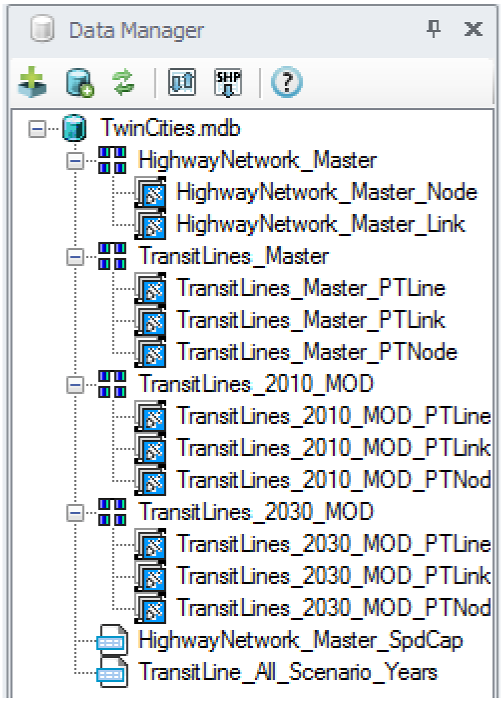
FILE NAME	FEATURE/TABLE	DESCRIPTION
TwinCities.mdb	HighwayNetwork_Master	Highway master network
	TransitLines_Master	Transit Master network
	TransitLines_2010_MOD	Updated transit lines in Year 2010
	TransitLines_2030_MOD	Updated transit lines in Year 2030
	TransitLine_All_Scenario_Years	Information to activate transit lines for each scenario year
SUPPORTLINK_MODE1_PT_2010.NTL		Walk access/egree links (Mode 1) for Year 2010
SUPPORTLINK_MODE2_PT_2010.NTL		Drive access links (Mode 2) for Year 2010
SUPPORTLINK_MODE4_PT_2010.NTL		Stop-to-Stop Transfer links (Mode 4) for Year 2010
SUPPORTLINK_MODE1_PT_2030.NTL		Walk access/egree links (Mode 1) for Year 2030
SUPPORTLINK_MODE2_PT_2030.NTL		Drive access links (Mode 2) for Year 2030
SUPPORTLINK_MODE4_PT_2030.NTL		Stop-to-Stop Transfer links (Mode 4) for Year 2030

All the non-transit access/egress links are directly connected between traffic analysis zone centroids and stops/stations. The drive-access links are also connected directly to the stops/stations, without utilizing any parking nodes. Table 2 lists these non-transit input files in addition to the transit data in the geodatabase file (e.g. 'TwinCities.mdb').

As shown in Figure 3, the geodatabase file contains various feature dataset such as highway master network, transit master network, and scenario-based transit network and standalone table. It is noted that the names for feature dataset or feature classes should not be changed, in order to properly utilize the PT model. These names are described as follows:

1) 'HighwayNetwork_Master' - highway master network
2) 'TransitLines_Master' - transit master network
3) 'TransitLines_2010_MOD' - update of transit network in Year 2010
4) 'TransitLines_2030_MOD' - update of transit network in Year 2030
5) 'TransitLine_All_Scenario_Years' - list of transit lines to activate the transit services

Figure 3. Transit Network Data in Geodatabase



Scenario-based highway and transit networks are created using the respective master networks. Since the highway network can include any user-defined link attributes related to the scenario years, only one feature dataset is required to create each highway network for each scenario year. However, unlike the highway network the transit network feature dataset cannot contain any user-defined attributes in addition to built-in line attributes. In particular, there is no built-in mechanism to re-route a line within a master transit network dataset in a future year. Instead, additional transit line data may be stored in the feature dataset 'TransitLines_2010_MOD' to manage scenario-specific transit networks, especially when the routing of existing transit lines in the master transit network will be changed in future years.

In the transit master network process, transit line names are used to activate the lines for each scenario year, as shown in Table 3. Note that the transit master network is constructed by including all the transit lines for the base year and all applicable future years. The transit master network process will activate only the transit lines listed for each scenario in the table.

Table 3. Information to Activate Transit Lines

OBJECTID	LINES_2010	LINES_2015	LINES_2020	LINES_2025	LINES_2030
1	2_W	2_W	2_W	2_W	3_1
2	2_E	2_E	2_E	2_E	3B_1
3	3_1	3_1	3_1	3_1	3_2
4	3B_1	3B_1	3B_1	3B_1	3_3
5	3_2	3_2	3_2	3_2	3_4
6	3_3	3_3	3_3	3_3	3_5
7	3_4	3_4	3_4	3_4	3B_2
8	3_5	3_5	3_5	3_5	3B_3
9	3B_2	3B_2	3B_2	3B_2	3B_4
10	3B_3	3B_3	3B_3	3B_3	3B_5
11	3B_4	3B_4	3B_4	3B_4	3E
12	3B_5	3B_5	3B_5	3B_5	3U_1
13	3C_1	3C_1	3C_1	3C_1	3U_2

3. Transit Fares

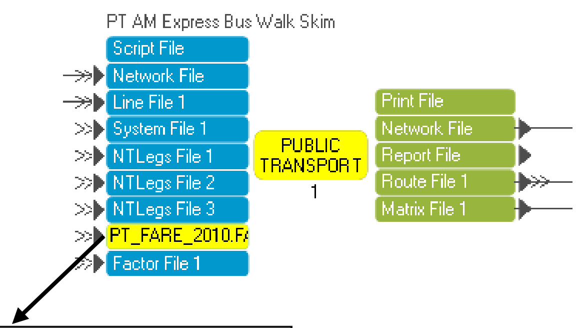
Transit fares are defined using an independent transit fare file in the PT program as shown in Figure 3. The PT analysis application utilizes fixed flat fares for transit modes 5,6,7 and 8 as is consistent with the previously adopted TRNBUILD model.

Table 4 and Figure 4 describe the settings to define the fare system in the PT model. All fares have been scaled back to year 2000 dollars for consistency with model calibration. In the peak period, flat fares are defined as $1.87 for local bus, local limited stop bus, and light rail and $2.49 for express. In off-peak period, these fares are reduced to $1.45 and $1.87, respectively. The fares for the commuter rail are based on the station-to-station services represented by the fare zone option.

Table 4. Transit Fares for Peak and Off-Peak

TRANSIT MODE		PEAK	OFF-PEAK
NUMBER	DESCRIPTION		
5	Local Bus	$1.87	$1.45
6	Local Limited Stop Bus	$1.87	$1.45
7	Express Bus	$2.49	$1.87
8	Light Rail	$1.87	$1.45
9	Commuter Rail	by Station-to-Station	

Figure 4. Settings of Transit Fares




```
FARESYSTEM NUMBER=1,
NAME="MODE 5,6 & 8-AM",
LONGNAME="FLAT FARE FOR MODES 5,6 AND 8 FOR AM PERIOD",
STRUCTURE=FLAT, SAME=SEPARATE,
IBOARDFARE=2.25

FARESYSTEM NUMBER=2,
NAME="MODE 7-AM",
LONGNAME="FLAT FARE FOR MODE 7 FOR AM PERIOD",
STRUCTURE=FLAT, SAME=SEPARATE,
IBOARDFARE=3.00

FARESYSTEM NUMBER=3,
NAME="MODE 5,6 & 8-OP",
LONGNAME="FLAT FARE FOR MODES 5,6 AND 8 FOR OP PERIOD",
STRUCTURE=FLAT, SAME=SEPARATE,
IBOARDFARE=1.75

FARESYSTEM NUMBER=4,
NAME="MODE 7-OP",
LONGNAME="FLAT FARE FOR MODE 7 FOR OP PERIOD",
STRUCTURE=FLAT, SAME=SEPARATE,
IBOARDFARE=2.25

FARESYSTEM NUMBER=5,
NAME="MODE 9",
LONGNAME="FAREZONE FROM-TO FARES FOR CRT",
STRUCTURE=FROMTO, SAME=CUMULATIVE,
FAREMATRIX=FMI.1.1, FAREZONES=NI.FAREZONE
```

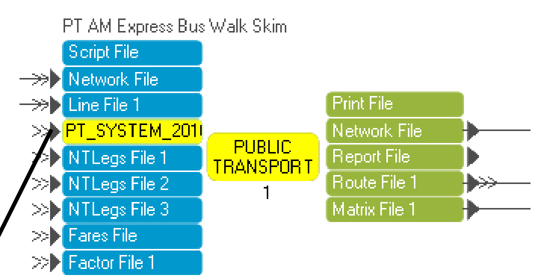
4. Transit Wait Curves

In TRNBUILD, a line's wait time is calculated as one-half of its headway, whereas PT uses wait "curves" to compute initial and transfer wait times at stop nodes based on the frequency of services, as shown in Table 5. These values are set in the PT module as X-axis for frequency and Y-axis for wait time as shown in Figure 5. For example, the initial wait time is 12 min for 27 min of frequency while the transfer wait time is 15 min for 20 min of frequency. In addition, two wait curves may be allocated to each stop node by 'IWAITCURVE' and 'XWAITCURVE' key words in the factor file.

Table 5. Transit Wait Time

INITIAL WAIT (min)		TRANSFER WAIT (min)	
FREQUENCY	WAIT	FREQUENCY	WAIT
1	0.5	1	0.5
16	8	4	8
27	12	12	12
48	15	20	15
160	20	40	20
		60	30

Figure 5. Settings of Transit Wait Curves




```
WAITCRVDEF NUMBER=1 LONGNAME="InitialWait" NAME="IWAIT" ,
           CURVE=1-0.5,16-8,27-12,48-15,160-20
WAITCRVDEF NUMBER=2 LONGNAME="TransferWait" NAME="XWAIT" ,
           CURVE=1-0.5,4-2,12-7.2,20-10,40-20,60-30
```

TRANSIT PARAMETERS

During transit path-building procedures, various parameters are introduced to enumerate and evaluate reasonable transit routes. Table 6 lists the parameter values used in transit path building and Figure 6 also shows the parameter settings in the input factors file. The parameter values have been converted from TRNBUILD settings to implement equivalent processes in PT.

Table 6. Transit Parameters in Factor File

PARAMETER	DESCRIPTION	WALK-ACCESS				DRIVE-ACCESS			
		BUS	EXP BUS	LRT	CRT	BUS	EXP BUS	LRT	CRT
Global Settings									
RECOSTMAX	Maximum weighted cost/time for any path to be either considered the minimum cost path or enumerated	290	290	290	290	290	290	290	290
SPREADFUNC	Function that computes SPREAD defining an upper cost limit for routes	2	2	2	2	2	2	2	2
SPREADFACT	Multiplicative factor used in multirouting function to compute SPREAD	1	1	1	1	1	1	1	1
SPREADCONST	Constant used in multirouting function to compute SPREAD	5	5	5	5	5	5	5	5
Modes									
DELACCESS MODE	Specifies which modes may not be used as access connectors	2	2	2	2	1	1	1	1
DELEGRESS MODE	Specifies which modes may not be used as egress connectors	2	2	2	2	2	2	2	2
DELMODE	Specifies modes that the program will not use during route enumeration	7, 8, 9	9	7, 9		7, 8, 9	9	7, 9	
Fare and Wait Times									
FARESYSTEM	Number of fare model that will apply to operators or mode (we used operator in our model, not mode).	1 to 4	1 to 4	1 to 4	1 to 4	1 to 4	1 to 4	1 to 4	1 to 4
REWAITMIN	Minimum weighted wait time for a leg bundle	2	2	2	2	2	2	2	2
REWAITMAX	Maximum weighted wait time for any leg bundle	15	15	15	15	15	15	15	15
IWAITCURVE	The number of wait curve to be used to calculate initial wait time (defined in system file)	1	1	1	1	1	1	1	1
XWAITCURVE	The number of wait curve to be used to calculate transfer wait time (defined in system file)	2	2	2	2	2	2	2	2
WAITFACTOR	Node specific wait time weighing factor (unitless factor)	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
Run Factors									
RUNFACTOR[1]	Mode specific weighing factor applied to transit in-vehicle time or non-transit leg time, in our case mode-1 is walk-access	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
RUNFACTOR[2]	mode-2 = auto-access	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
RUNFACTOR[3]	not used	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
RUNFACTOR[4]	mode-4 = transfer legs	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
RUNFACTOR[5]	mode-5 = Local bus	1.0	1.2	1.2	1.2	1.0	1.2	1.2	1.2
RUNFACTOR[6]	mode-6 = Local limited stop bus	1.0	1.2	1.2	1.2	1.0	1.2	1.2	1.2
RUNFACTOR[7]	mode-7 = Experss bus	1.0	1.0	1.0	1.2	1.0	1.0	1.0	1.2
RUNFACTOR[8]	mode-8 = Light rail	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
RUNFACTOR[9]	mode-9 = Commuter rail	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Choice Settings									
CHOICECUT	Eliminates alternatives with low probabilities of use at walk or alternative-alighting decision points	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05

The 'RECOSTMATX' keyword is set to limit the upper cost during the route enumeration. It is equivalent to 'MAXTHPATHTIME' in TRNBUILD.

PT can optionally implement a multi-routing process using the spread commands. The 'SPREADFUNC' specifies the function to compute the upper limit, the 'SPREADFACT' is the factor to be multiplied with the minimum cost, and the "SPREADCONST' is the constant value to be added to the minimum cost. In

TRNBUILD, the ‘COMBINE MAXDIFF=5’ is set to combine any available routes not exceeding best time plus 5 with the best line. The PT model replicates this setting using the spread function type 2 specified by ‘SPREAD= GCost(MinRoute)*SPREADFACT+SPREADCONST’ which is defined with ‘SPREADFACT=1’ and ‘SPREADCONST=5’.

Non-transit and transit modes can be restricted from inclusion in path-building using ‘DELACCESSMODE’, ‘DELEGRESSMODE’, and ‘DELMODE’. In the TRNBUILD program, ‘ACCESSMODES’ and ‘SKIPMODES’ were utilized as the equivalent commands.

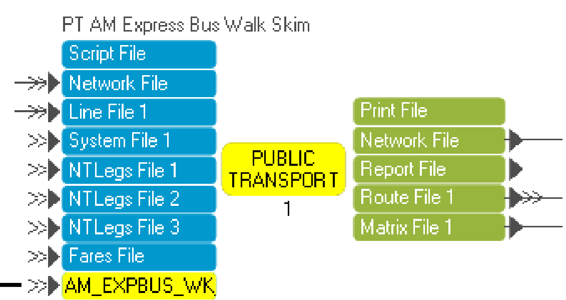
The wait time in PT can be calculated by setting the wait time curves as described in the previous section. Along with the wait curve settings, the wait time is also limited by maximum and minimum wait times using ‘REWAITMAX’ and ‘REWAITMIN’, similar to ‘IWAITMIN’, ‘IWAITMAX’, ‘XWAITMAX’, and ‘XWAITMIN’ in the TRNBUILD model. In addition, the ‘WAITFACTOR’ can be applied to specific nodes to weight the wait time. This corresponds to ‘IWAITFAC’ and ‘XWAITFAC’ in TRNBUILD.

Similar to ‘MODEFAC’ in TRNBUILD, the ‘RUNFACTOR’ keyword in PT is used to establish a hierarchy between the modes to define which mode is the “primary mode” and which modes act as secondary transfer modes. For example, since the path-building by the express bus (Mode=7) should use the express bus mode as the primary mode, ‘RUNFACTOR’ values for other bus modes such as local bus (Mode=5) and local limited stop bus (Mode=6) are set with 1.2 as the secondary modes.

The fare control statement, ‘FARESYSTEM’ can be defined with the number of the fare model that applies to the modes or operators. As in TRNBUILD, the PT model also uses the same flat fares for all the transit modes. The fare system numbers are set with 1 and 2 for the peak period and 3 and 4 for the off-peak period.

Figure 6. Settings of Transit Parameters

```
/*For Route Enumeration*/

RECOSTMAX=290      ; 290.0
SPREADFUNC=2       ; SPREAD function
SPREADFACT=1.0     ; SPREAD factor
SPREADCONST=5.0    ; SPREAD  constant

RUNFACTOR[1]  = 2.20     ; Walk Access
RUNFACTOR[2]  = 2.20     ; Drive Access
RUNFACTOR[3]  = 2.20     ; Station Access
RUNFACTOR[4]  = 2.20     ; Transfer
RUNFACTOR[5]  = 1.20     ; Local Bus
RUNFACTOR[6]  = 1.20     ; Local Limited Stop Buses
RUNFACTOR[7]  = 1.00     ; Express Bus
RUNFACTOR[8]  = 1.00     ; Light Rail
RUNFACTOR[9]  = 1.00     ; Commuter Rail

REWAITMIN=2
REWAITMAX=15

FARESYSTEM=1 MODE=5,6,8
FARESYSTEM=2 MODE=7,9

DELACCESSMODE = 2,3
DELEGRESSMODE = 2,3
DELMODE=9

/*For Route Evaluation*/
CHOICECUT=0.05

IWAITCURVE=1,  NODES=1633-99999
XWAITCURVE=2,  NODES=1633-99999
WAITFACTOR=2.2 NODES=1633-99999
```



TRANSIT PATH-BUILDING

The transit path-building procedure is used to accumulate impedances for the transit modes that are available within the mode choice model. The impedances include transit in-vehicle time and various out-of-vehicle time measures such as walk time and wait time. The path-building procedures also estimate transit fares for each mode. These impedance values are accumulated in matrix files based on definition of the mode choice model variables. It should be noted that transit paths are established by time period for each "access submode/line-haul mode combination" and that paths are developed based upon the multi-routing process.

As listed in Table 7, a total of 8 transit path building processes are performed for each time period. These access/line-haul mode combinations include:

– Walk-access and auto-access for local and limited stop buses (modes 5 and 6)
– Walk-access and auto-access for express bus (mode 7)
– Walk-access and auto-access for light rail (mode 8)
– Walk-access and auto-access for commuter rail (mode 9)

Table 7. Transit Skimming Matrices

<table>
<tr><th rowspan="2">NAME</th><th rowspan="2">DESCRIPTIONS</th><th colspan="4">WALK-ACCESS</th><th colspan="4">DRIVE-ACCESS</th></tr>
<tr><th>LOCAL & LIMITED BUS (MODES 5-6)</th><th>EXPRESS BUS (MODE 7)</th><th>LRT (MODE 8)</th><th>CRT (MODE 9)</th><th>LOCAL & LIMITED BUS (MODES 5-6)</th><th>EXPRESS BUS (MODE 7)</th><th>LRT (MODE 8)</th><th>CRT (MODE 9)</th></tr>
<tr><td>IVTT</td><td>In-vehicle time (minutes)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>WAIT1</td><td>Initial wait time (minutes)</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td></tr>
<tr><td>WAIT2</td><td>Transfer time (mode-4, in minutes)</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td></tr>
<tr><td>XFERS</td><td>Number of boadings</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>WALKT</td><td>Walk access & transfer Time (minutes)</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>FARE</td><td>Maximum Transit Fare ($)</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td><td>6</td></tr>
<tr><td>FIRST_MODE</td><td>First transit mode used</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td>1ST_MODE5</td><td>First node of transit mode 5 (Local bus)</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>1ST_MODE6</td><td>First node of transit mode 6 (Local Lim'd Bus)</td><td>9</td><td>9</td><td>9</td><td>9</td><td>9</td><td>9</td><td>9</td><td>9</td></tr>
<tr><td>1ST_MODE7</td><td>First node of transit mode 7 (Express bus)</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>LAST_MODE5</td><td>Last node of transit mode 5 (Local bus)</td><td>11</td><td>11</td><td>11</td><td>11</td><td>11</td><td>11</td><td>11</td><td>11</td></tr>
<tr><td>LAST_MODE6</td><td>Last node of transit mode 6 (Local Lim'd Bus)</td><td>12</td><td>12</td><td>12</td><td>12</td><td>12</td><td>12</td><td>12</td><td>12</td></tr>
<tr><td>LAST_MODE7</td><td>Last node of transit mode 7 (Express bus)</td><td>13</td><td>13</td><td>13</td><td>13</td><td>13</td><td>13</td><td>13</td><td>13</td></tr>
<tr><td>EXP_IVT</td><td>Mode-7 time (minutes)</td><td></td><td>14</td><td>14</td><td>14</td><td></td><td>16</td><td>16</td><td>16</td></tr>
<tr><td>LRT_IVT</td><td>Mode-8 time (minutes)</td><td></td><td>15</td><td>15</td><td>15</td><td></td><td>17</td><td>17</td><td>17</td></tr>
<tr><td>CR_IVT</td><td>Mode-9 time (minutes)</td><td></td><td></td><td></td><td>16</td><td></td><td></td><td></td><td>18</td></tr>
<tr><td>CRDIST</td><td>Mode-9 distance (miles)</td><td></td><td></td><td></td><td>17</td><td></td><td></td><td></td><td>19</td></tr>
<tr><td>DR_ACCT</td><td>Drive access time (minutes)</td><td></td><td></td><td></td><td></td><td>14</td><td>14</td><td>14</td><td>14</td></tr>
<tr><td>DR_ACCD</td><td>Drive access distance (miles)</td><td></td><td></td><td></td><td></td><td>15</td><td>15</td><td>15</td><td>15</td></tr>
</table>

The transit skim matrices generally consist of the following information:

1) In-Vehicle Travel Time (IVTT) – is time spent inside transit vehicles, in minutes. If a path between an origin zone and a destination zone includes several transfers, the IVTT will be the summation

of time spent inside all those transit vehicles. For example, if a transit path-building generates a path that includes riding a bus for 5 minutes, and then transferring to a train for 20 minutes, and finally transferring to another bus for 10 minutes, the IVTT for this origin-destination pair is 35 minutes.

2) Initial Wait Time – is time spent waiting to board a transit vehicle (or the first transit vehicle if the trip involves several transfers), in minutes.

3) Transfer Wait Time – is time spent waiting to transfer between transit vehicles, in minutes.

4) Boardings – is the number of boardings used by the “attractive” routes between zone pairs.

5) Walk Time – is the total time for walk access and walk transfer.

6) Maximum Transit Fare – is the maximum transit fare, in dollars, among the utilized transit modes.

7) First Mode – is the first transit mode used.

8) First Stop – is the first node of each transit mode.

9) Last Stop – is the last node of each transit mode.

10) Time by Mode – is time spent on a specific transit mode, in minutes.

11) Distance by Mode – is distance travelled on a specific transit mode, in miles.

12) Driving Time – is time spent driving to access a parking facility, in minutes.

13) Driving Distance – is distance traveled to access a parking facility, in miles.

TRANSIT ASSIGNMENT

The transit assignment process is used to distribute transit passengers to transit lines in both peak and off-peak periods. For each time period, the assignment process is performed separately for walk-access and drive-access transit modes.

In the PT program, the route files generated during the path-build process are directly utilized in the assignment process without the need to reset all the input files. Therefore, the parameters used for controlling the assignment process are identical to those used in the transit path-building process. The only difference is that in the path-building process, skims are set as the outputs, while in the assignment process, transit trips are added as inputs, and transit volumes by link by line are added as output.

SUMMARY RESULTS OF TRANSIT RIDERSHIP

The PT analysis application contains a process that generates a ridership summary by time period, access mode, and other major transit system components. It also provides a daily ridership summary by transit lines, line groups, transit modes, and operators.

Table 8. Transit Ridership by Transit Line

TRN_NO	NAME	LONGNAME	MODE	OPERATOR	USERA5	PKWK_ON	PKDR_ON	OPWK_ON	OPDR_ON	TOTAL_ON
1	10C	LMRP41CE13	5	3	10	144.63	20.77	244.67	15.43	425.5
2	10H	LMRP51CE00	5	3	10	12.63	1.79	7.42	0.45	22.29
3	10N	LMRPNOTW11	5	3	10	194.28	13.64	184.31	7.74	399.97
4	10_1	41CELMRP53	5	3	10	132.7	40.82	877.25	20.68	1071.45
5	10_2	51CELMRP00	5	3	10	42.39	10.3	0	0	52.69
6	10_3	LMRPNOTW00_10	5	3	10	39.43	2.36	193.48	8.02	243.29
7	10_4	LMRPNOTW10_10	5	3	10	157.91	9.45	17.66	0.7	185.72
8	10_5	NOTWLMRP50	5	3	10	243.62	40.11	558.76	43.88	886.37
9	10_6	NOTWLMRP51	5	3	10	246.75	42.5	561.84	45.5	896.59
10	111_1	254T66CE50	5	3	111	0	0	6.61	0	6.61
11	111_2	66CEOAWA10	5	3	111	76.47	0.07	0	0	76.54
12	113A	254T48GR00	5	3	113	0	0	6.66	0.03	6.69
13	113B	254T56LY51	5	3	113	0	0	3.61	0.01	3.62
14	113_1	48GR254T00	5	3	113	0	0	23.28	0.04	23.32
15	113_2	48GROAWA00	5	3	113	0	0	10.64	0.02	10.66
16	113_3	58LYOAWA00	5	3	113	80.17	0.22	63.02	0.08	143.49
17	114A	254T36LY00	5	3	114	0	0	8.97	0.09	9.06
18	114_1	36LYOAWA00	5	3	114	115.17	0.39	74.77	0.22	190.55
19	114_2	EXQUOAWA00	5	3	114	55.9	0.1	27.66	0.03	83.69
20	118	41CEOAWA00	5	3	118	56.83	10.59	12.14	0.5	80.06
21	11A	41CE46NI50	5	3	11	577.2	40.88	807.94	28	1454.02
22	11C	46NI41CE00	5	3	11	548.31	35.8	906.91	13.45	1504.47
23	11_N	46NI3SNI00	5	3	11	593.79	39.54	892.38	12.91	1538.62
24	11_S	46NI3SNI00	5	3	11	493.99	34.89	806.91	26.96	1362.75
25	121_1	BLHASPSC10	5	9	121	25.67	4.9	50.02	2.13	82.72

Table 9 summarizes the transit ridership for each transit mode.

Table 9. Transit Ridership by Transit Mode

MODE	PKWK_ON	PKDR_ON	OPWK_ON	OPDR_ON	DAILY_ON
5	60247.15	10577.62	75364.69	3421.67	149611.13
6	9624.42	1073.17	10266.66	718.69	21682.94
7	7981.62	5231.74	6035.31	611.73	19860.4
8	5123	1711.28	3850.34	1434.94	12119.56
9	2.46	12.83	0	0	15.29

Table 10 shows the summary of transit ridership for each transit line group (referred as 'USERA5' in the PT line data).

Table 10. Transit Ridership by Transit Line Group

LINE_GRP	PKWK_ON	PKDR_ON	OPWK_ON	OPDR_ON	DAILY_ON
2	4141.01	841.74	3946.39	203.63	9132.77
3	4369.02	1632.9	5422.15	296.01	11720.08
4	4074.51	680.95	4763.83	162.52	9681.81
5	2218.55	81.38	6435.81	177.92	8913.66
6	2233.14	469.16	2662.12	107.47	5471.89
7	731.1	175.22	743.99	25.1	1675.41
8	52.02	0.32	114.5	2.96	169.8
9	499.27	17.16	1186.98	104.78	1808.19
10	1214.34	181.74	2645.39	142.4	4183.87
11	2213.29	151.11	3414.14	81.32	5859.86
12	1083.69	224.94	617.65	28.7	1954.98
14	1415.15	105.71	3691.36	174.47	5386.69

Table 11 summarizes the transit ridership for each transit operator.

Table 11. Transit Ridership by Transit Operator

OPERATOR	PKWK_ON	PKDR_ON	OPWK_ON	OPDR_ON	DAILY_ON
1	83.29	154.63	0	0	237.92
2	348.07	331.02	43.92	8.43	731.44
3	78381.57	16525.58	92189.96	5880.71	192977.82
4	3061.63	966.33	2179.06	181.49	6388.51
5	333.37	148.25	82.08	3.36	567.06
6	18	25.92	0	0	43.92
8	434.01	368.27	515.84	49.84	1367.96
9	303.09	85.82	483.48	59.35	931.74

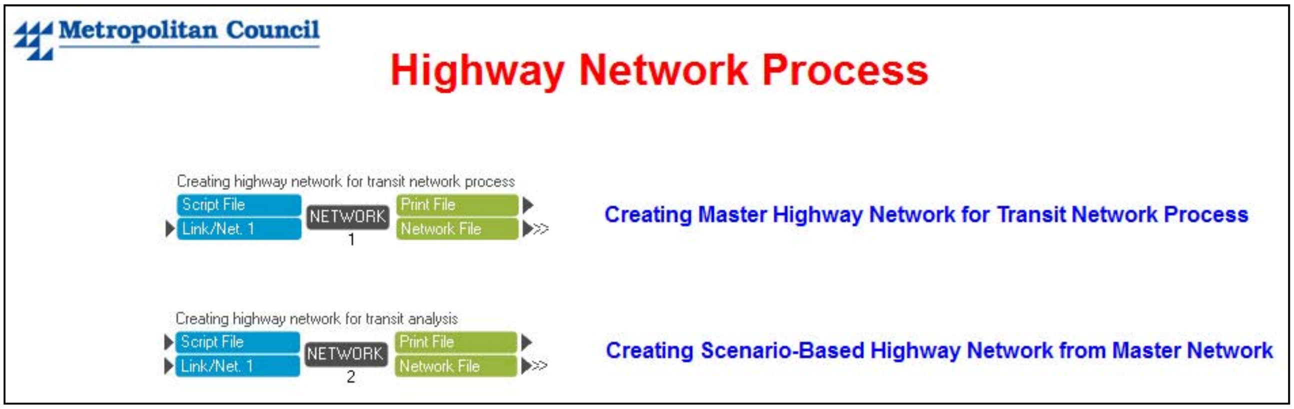
TRANSIT MODEL PROCESSING DETAIL

The PT model has been developed by converting the former TRNBUILD model. As noted previously, part of this effort includes the development of two new processes. One is the transit network process to generate a scenario-specific transit network from a transit master network. The other is a ridership summary process to accumulate the transit ridership by the transit components. This section describes each process in greater detail.

1. Highway Network Process

As shown in Figure 7, the highway network process generates two different highway networks from the master highway network contained in the geodatabase file. The first highway network is exactly same as the input master network, but transit time is added based on a link computation. This network is used to extract the master transit network from the geodatabase file in the following Cube application. The other highway network is the scenario-specific highway network that is used to implement both transit path-building and transit assignment for each scenario year.

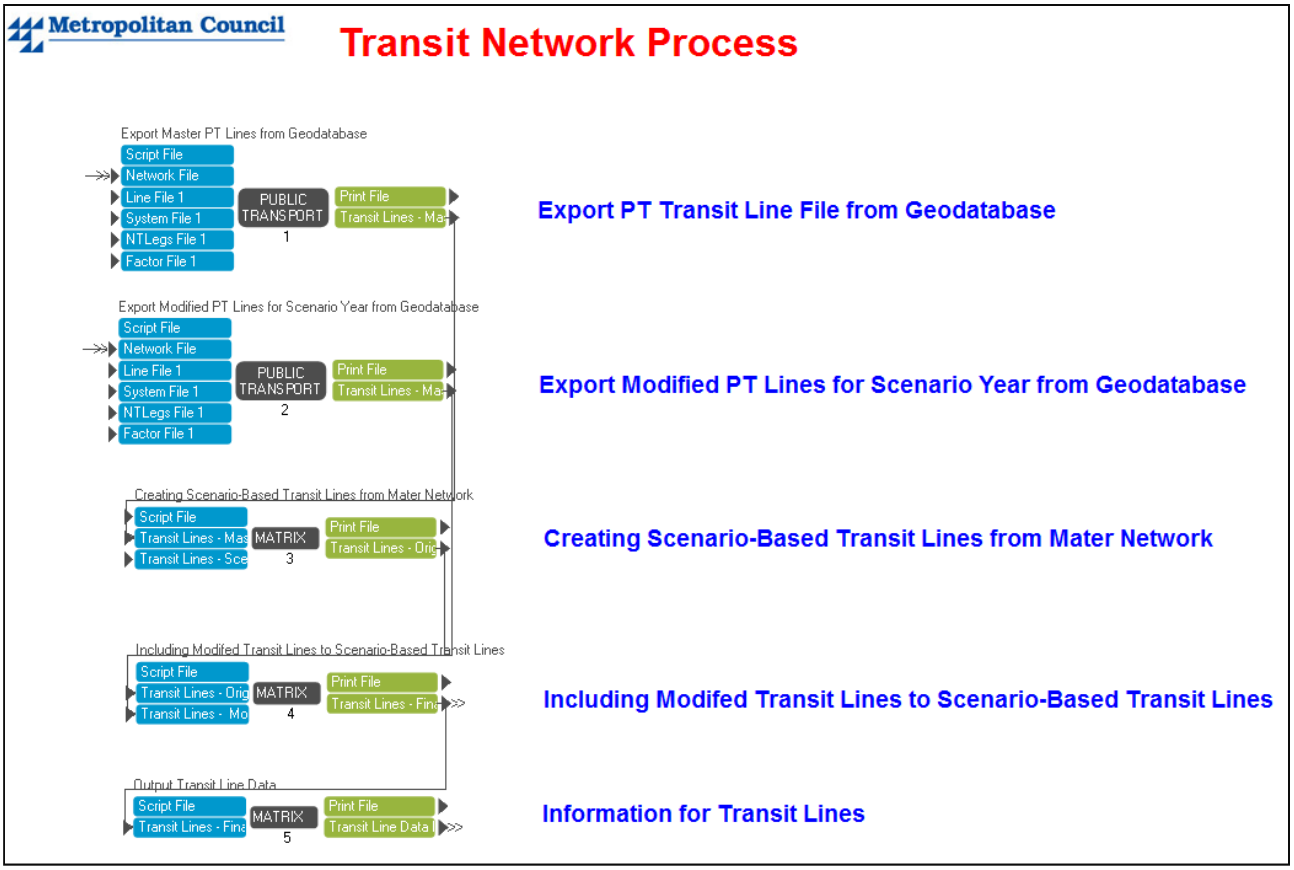
Figure 7. Highway Network Process



2. Transit Network Process

Figure 8 shows the transit network process to create the scenario-based transit network as well as the transit line information that is used to summarize the transit ridership as the post-process. The scenario-based transit network is created using both the transit line activation data and the modified transit line information for each scenario year.

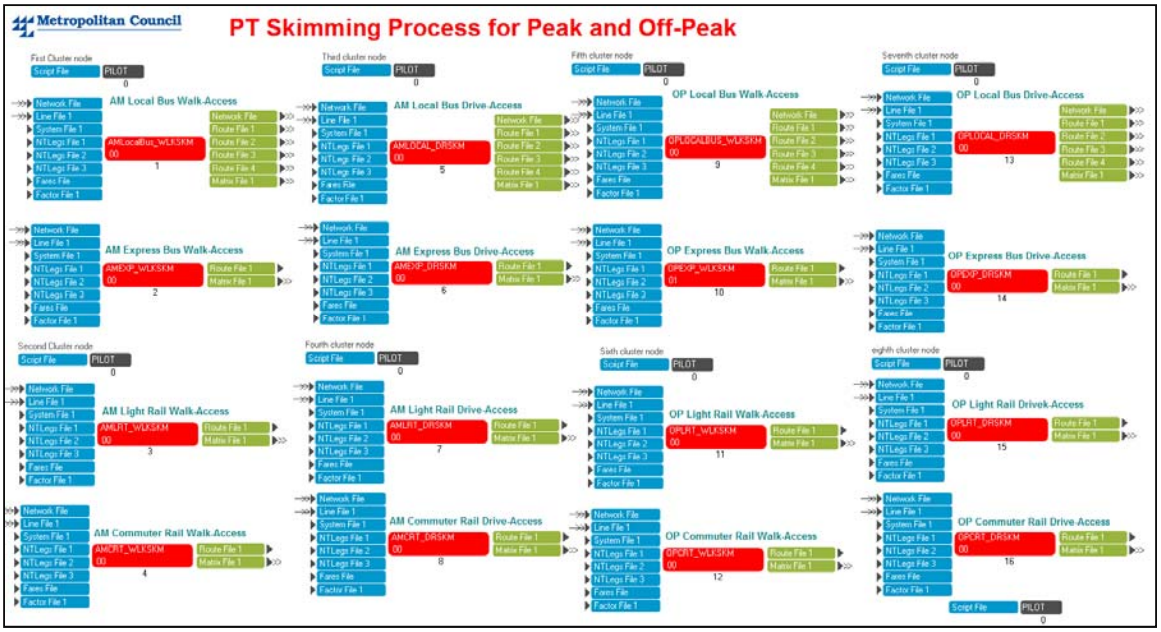
Figure 8. Transit Network Process



3. Transit Skimming Process

The transit skimming process includes 16 Cube applications by four major transit modes (e.g. local bus, express bus, light rail, and commuter rail), two access modes (e.g. walk-access and drive-access), and two time periods (e.g. peak and off-peak), as shown in Figure 9. Note that the skimming process for local bus generates 4 transit route files that are directly utilized in the transit assignment process. Since the PT program is based on the user class settings defined in each module, these settings should be identical between the transit path-building process and the transit assignment process.

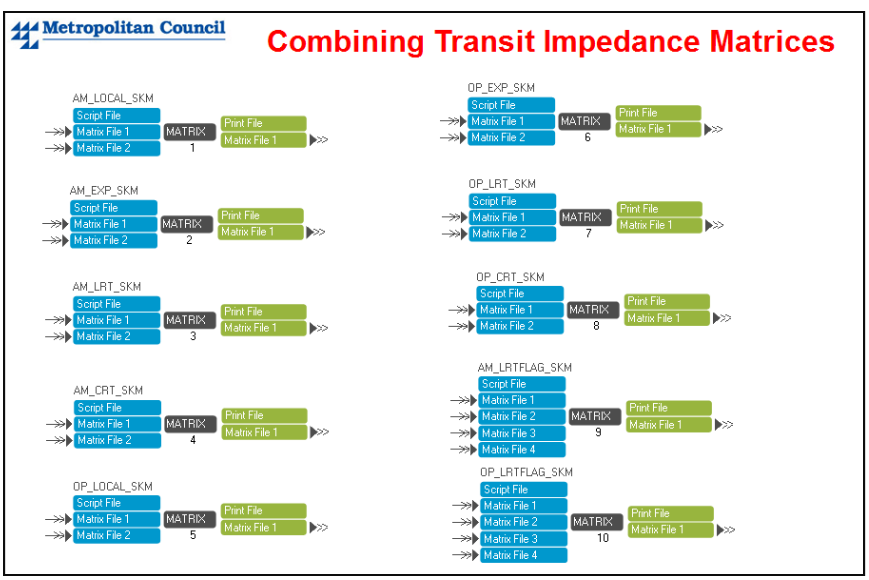
Figure 9. Transit Skimming Process



4. Combining Transit Impedance Matrices

In a transit skimming post-process, the skim matrices are combined to be used in the subsequent mode choice process. Figure 10 shows the combining process, which has been adopted from the TRNBBILD model currently used by the Metropolitan Council.

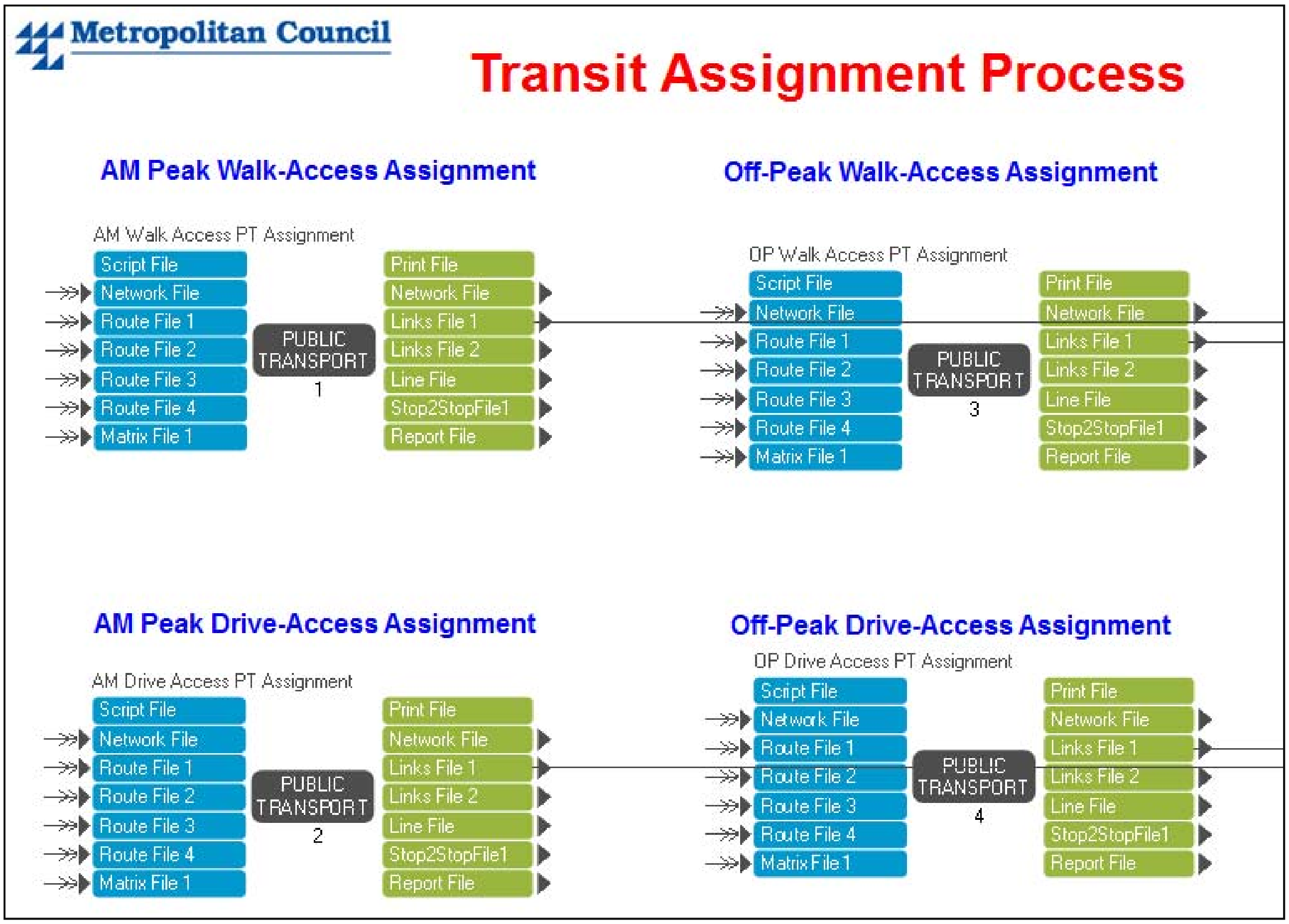
Figure 10. Transit Skimming Post-Process



5. Transit Assignment Process

Figure 11 shows the transit assignment process with four PT programs by non-transit access modes and time periods. The PT program can generate various useful estimates such as link results, line results, stop-to-stop results, etc.

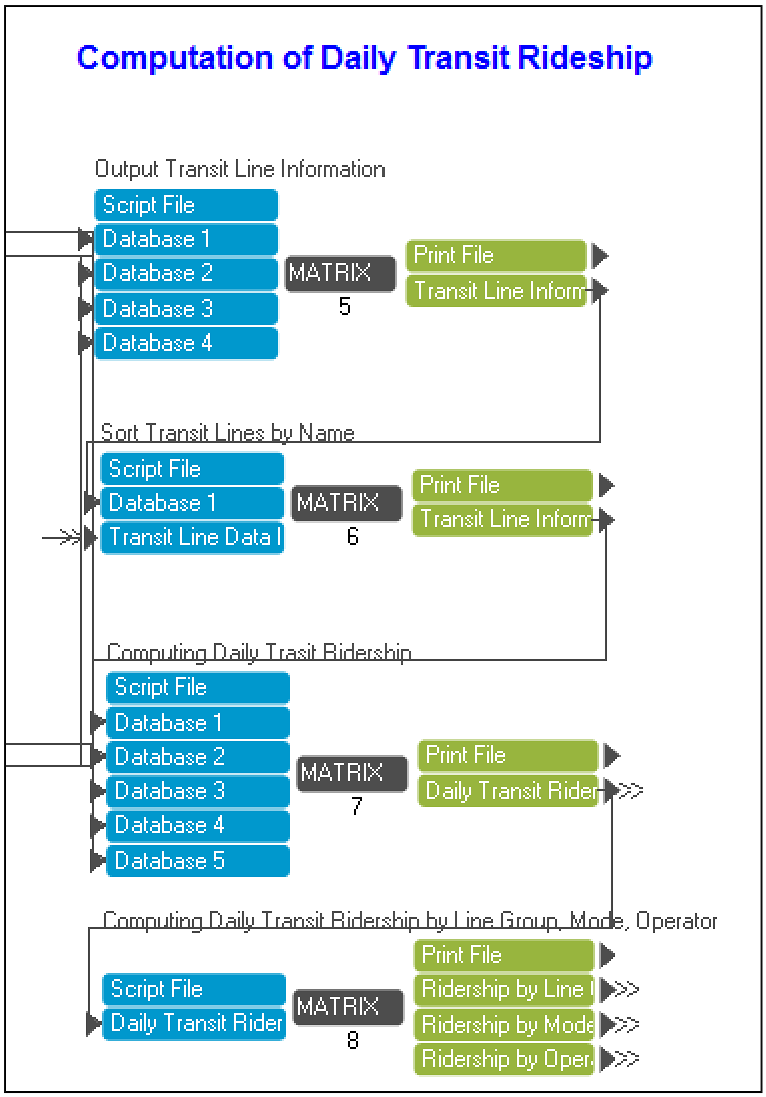
Figure 11. Transit Assignment Process



6. Summary of Transit Ridership

A transit summary post-process is finally performed using the link-based results, as shown in Figure 12. It summarizes the transit ridership by access mode and time period. Daily ridership is also summarized by transit line, transit line group, transit mode, and transit operator.

Figure 12. Transit Summary Process



Citilabs, Inc.
1211 Miccosukkee Road
Tallahassee, Florida 32308

www.citilabs.com