

VzCost Technical Documentation:

VzLoop Pre-Processing



Copyright 2002-2003 Verizon. All rights reserved.

Table of Contents

- 1.0 Executive Overview / Introduction..... 1
- 2.0 Process Details 5
 - 2.1 Verizon NW Data Systems/Data Download and Development 5
- 3.0 Table Development from AAIS 6
 - 3.1 AAIS Data Acquisition..... 6
 - 3.2 Bulk, Terminal, and DSL file descriptions and development..... 7
 - 3.2.1 Bulk file..... 7
 - 3.2.2 Terminal File..... 8
 - 3.2.3 DSL file - Inclusion of DSL lines 8
 - 3.2.4 Circuit file 9
 - 3.2.5 DS1 file 9
 - 3.3 Bulk and Terminal file modifications 10
 - 3.3.1 Removal of inappropriate Bulk file records..... 10
 - 3.3.2 Extracting and identifying key field data - Terminal ID..... 11
 - 3.3.3 Creation of Zip Code field 11
 - 3.3.4 Reconciliation of the Terminal/Bulk data and creation of interim table 11
 - 3.3.5 Preparation of table for improved Geo-coding results..... 12
 - 3.3.6 Geo-coding discussion and preparation of interim table for geo-coding 12
 - 3.4 Creation of Demand and Demand_value tables..... 13
 - 3.4.1 Demand table 14
 - 3.4.2 Demand table description..... 14
 - 3.4.3 Integration of non-switched/DS1 lines into the Demand table 16
 - 3.4.4 Demand_value table..... 16
 - 3.4.5 Demand_value table description..... 16
 - 3.5 Terminal_snap and Pre_network tables 17
 - 3.5.1 Development of Terminal_snap table 18
 - 3.5.2 Development of Pre_network table 19
 - 3.6 Summary of AAIS - Derived Tables 19
- 4.0 Network Input into the Pre-process 19
 - 4.1 Table Development from ICGS/Fieldview and ICAPS 19
 - 4.1.1 File Development 20
 - 4.1.2 ICGS, FieldView, ICAPS 20
 - 4.1.3 ICGS..... 20
 - 4.1.4 ICGS CPR file..... 21
 - 4.1.5 ICGS Compliment file 21
 - 4.1.6 FIELDVIEW 22
 - 4.1.7 ICAPS 22
 - 4.1.8 ICAPS WANET file structure 22
 - 4.1.9 NETCAPONLINE WADLC and WAXBOX file structure 23
- 5.0 Programs and Processes..... 23

5.1	Processes	24
5.1.1	Terminalsnap True Up	24
5.1.2	CIP Trueup	25
5.1.3	Final Trueup	25
5.2	Minor Programs	25
5.2.1	CleanCPR.exe	26
5.2.2	CalcRelxy.exe	26
5.2.3	CleanFamily.exe	26
5.2.4	SnaptoNet.qry	26
5.3	Major Programs	26
5.3.1	Termsnap.exe	27
5.3.2	Famatch.exe	32
5.3.3	Spannet.exe	36
6.0	Post Network table activities.....	37

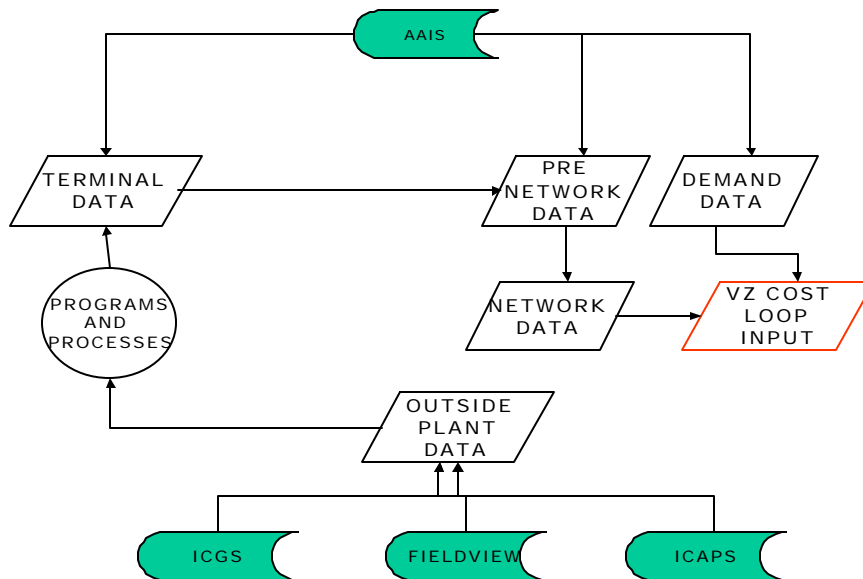
VERIZON NORTHWEST LOOP PRE-PROCESSING

1.0 Executive Overview / Introduction

This document explains the data gathering and processing activities, referred to as “pre-processing activities,” required to convert Verizon Northwest Inc.’s (“Verizon NW’s”) customer assignment and plant data into a format suitable for input into VzLoop.

In order to construct a modeled network that accurately identifies customer demand, Verizon made extensive use of ‘real world’ data systems, the systems used in the everyday management of Verizon NW’s outside plant (“OSP”) network. The Verizon systems used in the development process all contain either an address or some type of location coordinate system that allows for the relative ‘placement’ of major network components to provide the basis for a forward looking network model. In addition, the systems provide the developers with the information required to arrange the network based on relationships between the distribution terminals, SAIs (cross-connect boxes), DLCs and other network points.

The following flowchart summarizes the loop pre-processing activities, with data sources highlighted:



The pre-processing activities for the Verizon NW loop data consist of five major processes, each with many sub-processes. The five major processes are:

- 1) Data downloads
- 2) Cross-matching/reconciliation between terminal and customer files
- 3) Geocoding

- 4) Table development
- 5) Programs/processes

The **data download process** consists of the capture and download of selective data from a combination of Verizon Washington's legacy and OSP (Outside Plant) planning systems into various source tables.

The AAIS (Assignment Activation and Inventory Services) is a Verizon legacy system used in day-to-day facilities assignment and inventory management. For VzLoop's purposes, the AAIS provides customer and serving terminal information for switched access customers, DSL information, information on non-switched circuits, and DS1 circuit information. AAIS data are loaded into pre-process tables reflected in the above diagram as Terminal, Pre-Network, and Demand data.

The ICGS (Interactive Computer Graphics System), also a Verizon legacy system, provides OSP facility data including cable type, cable length, cable/component size, account code, etc. ICGS data is loaded into several tables represented by Outside Plant Data in the above diagram. A condensed version of the ICGS, called FIELDVIEW, is used to identify facility locations through the use of a built-in graphical interface. Data from FIELDVIEW's .dgn files are downloaded and captured in a series of files, also represented in the Outside Plant Data above.

ICAPS (Integrated Customer Access Planning System) is an OSP planning tool used to develop feeder facility routing based on existing feeder cable routes. The ICAPS is also used to identify quantities and locations of existing pair gain devices and cross-connect boxes. Data from the ICAPS is captured and placed into tables also represented in the Outside Plant Data.

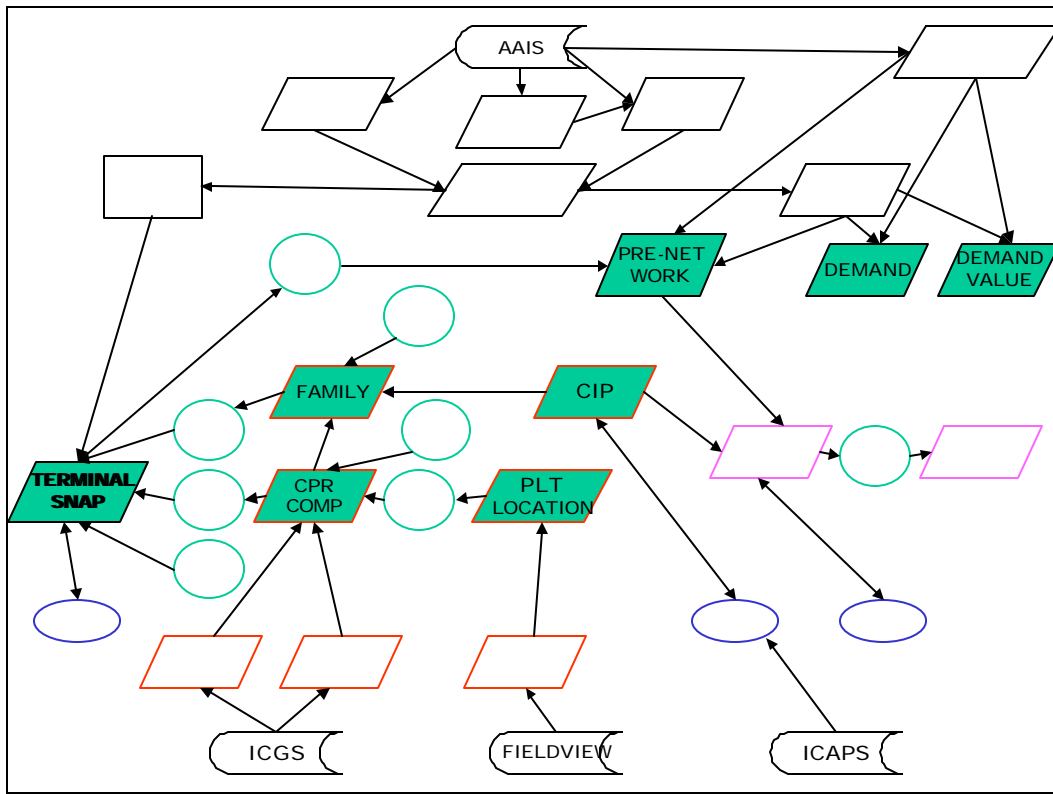
Location of customer terminals through geo-coding is one of the basis of the loop preprocess. In order to prepare data suitable for effective geo-coding, a **Terminal and Customer (Bulk) file cross match and reconciliation** is performed. At a high level, the reconciliation process copies the appropriate zip codes from the bulk table to records in the terminal table and incorporates logical changes to the data to produce a table capable of producing the most accurate geo-coding results possible with the available data. It also highlights discrepancies between the two tables. Following geo-coding of the Terminal table, the Terminal table is reviewed and made ready for table development. Detailed discussion of the cross match and reconciliation processes are discussed below.

Geo-coding provides one of the basic items required for an effective loop model by determining initial relational positions of existing customer serving terminals. The geo-coding process uses terminal street addresses to assign angular (latitude and longitude) coordinates to the terminal. Only customer serving terminals are geo-coded. Other network items, such as cross connect boxes and pair gain devices, are located using other methods.

As explained in later sections, varying degrees of accuracy exist in the geo-coded results, requiring, in some instances, further steps to improve the location accuracy.

Table development involves the creation of eight data tables to house the data from downloaded source tables. Tables used for pre-processing, their System sources, and representation in the above summary diagram are:

- Demand – AAIS : Demand data
- Demand Value - AAIS : Demand data
- Pre_network - AAIS: Pre-network data
- Terminal_snap – AAIS: Terminal data
- Plt_location – FIELDVIEW : Outside plant data
- Cprcomp – ICGS: Outside plant data
- CIP – ICAPS: Outside plant data
- Family – ICAPS and ICGS: Outside plant data



Tables used in the pre-process.

Referring to the above diagram and beginning with tables with an AAIS origin:
The Demand and Demand Value tables originate from the Bulk (Customer) file and provide customer demand information for the VzLoop executable program. The terminal_snap and pre_network tables are based on selected information from the

Terminal and Bulk tables (AAIS), respectively, and contain terminal-related information, including location and terminal pair count.

From ICGS, FIELDVIEW, and ICAPS: The Plt_Location table is a compilation of FIELDVIEW files and provides cable location information. The CRPCOMP table is developed from the CPR and Compliment tables (ICGS) and provides a 'one-stop' table for facility information contained in those tables, including account and cable pair count information. The CIP (Cross-connect, ICAPS, Pair-gain) table contains information from the ICAPS database that is used to position existing cross connect boxes, pair gains, and feeder routes in the network. The detailed activities and processes used to develop these are discussed in the Table Development section. The family table contains serving terminal grouping relationships and is developed from the CPRCOMP and CIP tables.

Following geo-coding and table development, all data is ready to be utilized in the network design programs and processes.

Programs/processes: The Verizon NW pre-processing procedure utilizes several programs and procedures to produce the Network table for input into VzLoop. Details on these programs and processes are discussed in the Programs/Processes section. At a high level, the program and process functions are as follows. Note that the programs and processes are discussed in the order they occur in the process. Note also that '**' designates the major programs in the process.

CleanFamily.exe - Removes undesired leading characters from the cable count field of the Family table.

CleanCPR.exe - Removes undesired leading characters from the cable count field of the CPRCOMP table.

CalcRelXY.exe – Converts the geo-coded latitude and longitude of each terminal in the Terminal-Snap file to a distance in feet (in x and y units) from the central office.

****Termsnap.exe** – Develops an x-y location for each terminal in the Terminal-Snap file based on cable and terminal count and location. Also determines information required to determine terminal construction type, i.e., aerial, buried, or underground..

Terminal_Snap table true up (pre FaMatch) - Process that allows developers to determine whether the original geo-coded or cable-derived locations should be retained.

****Famatch.exe** - Assigns the appropriate family (grouping) to each terminal, thereby accounting for existing demand grouping of terminals to DLCs, cross-connect boxes, and the central office.

Terminal_snap true up (post FaMatch) – Process that allows developers to regroup terminals to their appropriate family, if necessary.

CIP table true up - Ensures that the CIP table contains no control point inaccuracies. Also ensures that the CIP table contains a cross connect or pair gain record for each serving terminal family (grouping).

SnapToNet.sql - Copies all location coordinates, groupings, and terminal construction type information from the terminal_snap file to the appropriate pre_network file records.

Pre_network true up – Final check before running spannet.exe. Ensures that all data has transferred correctly to the pre-network and deletes records with an x,y location of 0,0.

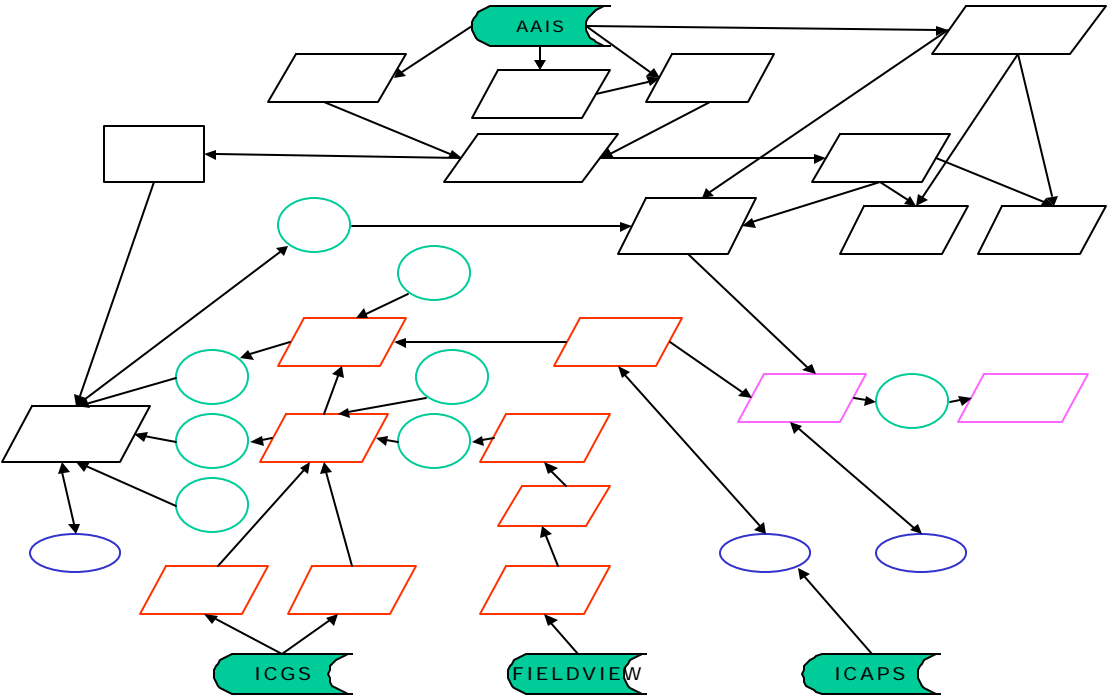
****Spannet.exe** - Connects the terminals, cross connects, pair gains, and control points to from a logical network as directed by the pre_network table. Outputs the network table for use by VzLoop.

Following the completion of Spannet.exe, pre-processing is complete. The network table contains all the network data and relationships in the proper format to input into the VzLoop system. The network table, along with the Demand and Demand Value tables, is delivered for input into the VzLoop process. Again, details on all the above processes are discussed in the remainder of this document.

2.0 Process Details

2.1 Verizon NW Data Systems/Data Download and Development

The Verizon NW pre-processing procedure uses three Verizon West systems for data input. Two of the three, AAIS and ICGS, are legacy systems used by network and assignment personnel on a daily basis. The third, ICAPS, is an OSP Planning tool used to track network demand patterns and facility utilizations. The FIELDVIEW system, also shown below, is a subset of the ICGS system. The following section describes in detail the derivation of data tables from the AAIS.



Preprocess flow chart. Source systems highlighted

3.0 Table Development from AAIS

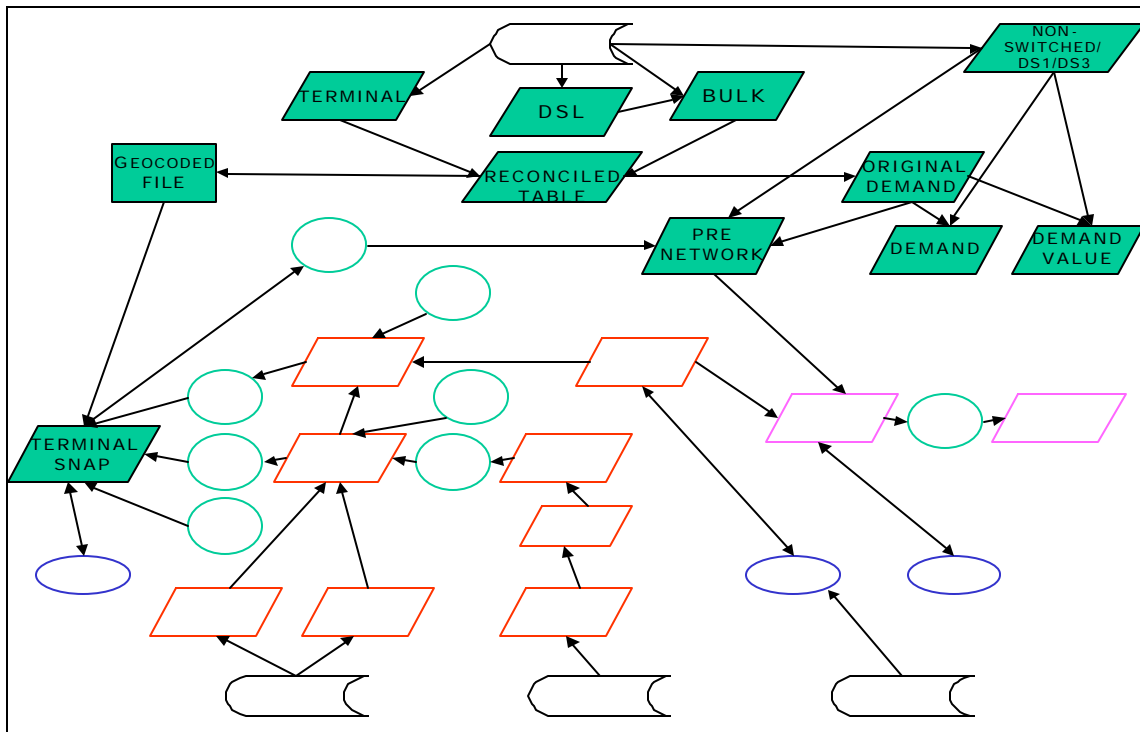
AAIS contains all facility assignment related data for Verizon NW customers. AAIS is composed of two parts- AAIS-Core and AAIS-DS. Included in the AAIS-Core portion of AAIS is switched customer data that include telephone number, service address, class of service, serving cable ID, pair number, DSL service, etc. AAIS-Core also includes serving terminal information, including terminal ID, terminal address, terminal type and size, etc, and information concerning non-switched circuits. AAIS-DS houses information regarding HiCap, i.e., DS1 circuits in the network.

AAIS data is the major player in the pre-process, providing the basis of the geocoded terminal information, VZLoop's Demand and Demand-Value tables, as well as the initial information contained in the Terminal_Snap and Pre_Network tables.

3.1 AAIS Data Acquisition

The AAIS-Core data acquisition process begins with a request for switched customer and serving terminal data, DSL information, and non-switched circuit information from the AAIS-Core group. The AAIS-Core personnel query the data from the AAIS database and provide the delimited text files (Terminal, DSL, Bulk, and Non-switched) shown below. DS1 information is received through a query written by the AAIS-DS group. As received from the AAIS groups, the files are delimited with various characters (characters such as comma, semicolon, '@', or tab separate the data fields), with multiple different delimiters per data row. Logic must be applied in the pre-process to properly extract the required data.

The information in the serving terminal and customer information files forms the basis for the Terminal and Bulk tables, respectively. The DSL information is added to the Bulk file later in the process, and the non-switched/DS1 information is added directly to the final tables (see flowchart).



AAIS – derived tables

3.2 Bulk, Terminal, and DSL file descriptions and development

3.2.1 Bulk file

The bulk file is one of the five files extracted directly from the AAIS. The bulk file serves as a warehouse for all customer specific information in the process, including telephone number, service address, service type, terminal ID, etc.

Bulk file structure

- Catalog & Wire Center
- Phone Number
- Line Equipment
- Switch Number (within wire center)
- Location Number (1=CO, 2 or higher = Remote)
- Facility path (begins at line equipment and ends at customer's premise) (includes miscellaneous equipment, Dmark (CLECs), cableID and pair, carrier, crossboxes & **servicing terminal**)
- Primary Class of Service
- Service type
- Status
- Address

Sample bulk file records: - Data fields separated by '|', serving terminal ID shown in **bold**, ZipCode in *italics*, service type underlined, placeholders for telephone numbers.

NW77224|360-555-1212|DUBS0120531|1|2|OPM1-196@AB + DAB-30@**952**|B1|B1|WKG|220 N. DUBOIS RD, ARIEL, WA 98603|
NW77224|360-555-1213|DUBS0120309|1|2|OPM1-110@AB + DAB-35@**952**|B1|B1|WKG|220 N. DUBOIS RD, ARIEL, WA 98603|
NW77224|360-555-1214|LCE0410422|1|1|3-1667@BA + DBA-779@**3311**|B1|B1|WKG|1635 DOWN RIVER DR, WOODLAND, WA 98674|
NW77224|360-555-1215|LCE0221516|1|1|2-1228@BA + DBA-653@**3001**|B1|B1|WKG|1620 DOWN RIVER DR, WOODLAND, WA 98674|
NW77224|360-555-1216|RENO010613|1|4|1DU1-206@AJ + DAJ-245@**228**|B1|B1|WKG|111 MERWIN HATCHERY CT, ARIEL, WA 98603|
NW77224|360-555-1217|LCE0421430|1|1|2-2019@BF + DBF-193@**2012**|B1|B1|WKG|160 PINKERTON DR, WOODLAND, WA 98674|

3.2.2 Terminal File

The terminal file is another of the five files extracted directly from the AAIS. It contains all information regarding serving terminals required by the pre-process, including terminal ID, terminal address, terminal type, size, count, etc.

Terminal file structure

Catalog & Wire Center

Lead

Terminal ID

Type

Capacity

Count (cable ID & pair ranges)

IPID

Address

Sample terminal file records. Data fields separated by '|'. Serving terminal ID shown in bold

NW77224|**952**|FC|25|DAB-26/50|220 N DUBOIS RD , (ARIEL, WA), SUB=N DUBOIS RD 6N2EF14|PED|
NW77224|**3311**|FC|50|DBA-776/800|13450|1635 DOWN RIVER DR, WOODLAND, WA|OBL|
NW77224|**3001**|FC|50|DBA-651/700|1620 DOWN RIVER DR , (WOODLAND, WA), SUB=SONOCO BLDG #2 MAP

3.2.3 DSL file - Inclusion of DSL lines

The information acquired from the AAIS DSL file consists of a listing of 9-digit telephone numbers with DSL service. Before any data modifications are done to the Bulk file, a match on the telephone numbers in the DSL file is performed against telephone numbers in the bulk file. Because ADSL does not require an additional cable pair for a line in the Bulk file,¹ the inclusion of DSL lines in the Bulk file does not

¹ All customers represented by records in the Bulk file have a telephone number. The addition of DSL to that telephone numbers does not require an additional pair.

increase the total number of lines (pairs) in the table, but merely causes a change in service type (See sample bulk file above for service types). When a match occurs, the service type in the bulk file is changed to 'DSL' if the initial service type indicated a business line, or changed to 'RESDSL' if the initial service type indicated a residential line.

3.2.4 Circuit file

The fourth file received from the AAIS is the non-switched, or circuit file. This file contains information concerning the low capacity (DS0 or lower) circuits that are not switched, and therefore, not contained in the Bulk file or the DS1 file.

Circuit file structure

AAIS Catalog/WC

Cable ID, Pair, serving terminal

Service type (CKT) – Contains CKT (circuit) records exclusively

Status (Working) – Contains status types representing working circuits exclusively

Circuit ID

Sample circuit file records (fields separated by '|')

```
NW77402|1-897@AG + DAG-50@395||CKT|WKG|74.HCGS.160903..PN|
NW77402|1-899@AG + DAG-3@395||CKT|WKG|86.HCGS.503009..GTEW|
NW77402|1-922@5667||CKT|WKG|86.HCGS.503290..GTEW|
NW77402|1-924@5667||CKT|WKG|86.HCGS.503689..GTEW|
```

3.2.5 DS1 file

The DS1 file is the fifth file received from the AAIS. It contains information about the DS1 circuits in Verizon's network.

DS1 file structure

State

Ckt Type

Rate Code

Service Type

Circuit Name

CLLI A

CLLI Z

End User Name

End User A Add

City

State

Zip

LSO A
End User Name
End User Z Add
City
State
Zip
LSO Z
Number of Riders

Sample DS1 records – Fields separated with ‘|’, end user name substituted with ‘xxxx’

WA|FIBER|DS1|UH|01/UHMA/003702/ /GTNW/ |CAMSWAXXDS1|VANCWA01DS0|xxxx|330 NE
5TH AVE, | CAMAS|Washington||CAMSWAXX|xxxx|4100 PLOMONDON, |
VANCOUVER|Washington||CAMSWAXX||

WA|FIBER|DS1|HC|74/HCGA/155940/ /PN / |BLLVWASH|BOTHWAAKW03|xxxx|14808 SW 16TH, |
BELLEVUE|Washington||BLLVWASH|xxxx|19807 NORTHCREEK PKWY, |
BOTHELL|Washington||BOTHWAAK||

3.3 Bulk and Terminal file modifications

Modifications to the downloaded Bulk and Terminal files are required to appropriately use their data in the pre-processing procedure. Certain data need to be shared between the files and both of the files contain data fields that need to be dissected in order to provide required data. To accomplish this, the developers decided to create a third file to contain the resultant data from the Bulk and Terminal files. What follows is a detailed description of the activities required to transform the data from the Bulk and Terminal files into data suitable for use in the pre-process procedure.

3.3.1 Removal of inappropriate Bulk file records

Records exist in the bulk file that exaggerate the number of lines (pairs) in use in the network. The initial Bulk file modification is the removal of these records.

1. Records with duplicate wire center and facility path.
Records exist in the Bulk file that represent the same cable pair. For example, a ‘Teen Line’ service utilizes the same cable pair as the R1 service at a particular location, but is listed as a separate record in the Bulk file. The duplicate record is removed from the Bulk file.
2. Records with a null facility path field
Records with a null facility path are not working lines and are removed.
3. Records with a null line equipment field
These type records are similar to records with a null facility path and are removed

4. Records with service types of 'None', 'test', or '****'

Records with these service types represent CO test or other entries in the bulk file that have no associated outside network

3.3.2 Extracting and identifying key field data - Terminal ID

The catalog/wire center and Terminal ID fields in the Terminal and Bulk files are defined as the key fields. As the key fields in this process, the catalog/wire center and terminal ID information is used to link the customer information in the Bulk file to the terminal information in the Terminal file. To link the files successfully, the terminal ID data in both files must be an exact match. As obtained from the AAIS group, the terminal ID information in the Bulk file is imbedded in the information beyond the last '@' in the facility path field (see bulk file sample records above) and must be separated from the remaining facility path information in order to create an exact match with the terminal ID information in the terminal file. The terminal information in the terminal file is defined separately as the Terminal ID field and no parsing is necessary. Once the Terminal ID information is separated in both files into a separate field, the process is ready to continue.

3.3.3 Creation of Zip Code field

In order to be accurately geo-coded, a terminal address needs to contain a Zip Code. A challenge exists in that the Terminal address data in the Terminal file do not contain a Zip Code. The Zip Code information for terminal geo-coding purposes must be obtained from the Bulk file.

3.3.4 Reconciliation of the Terminal/Bulk data and creation of interim table

In order to reconcile and match the Bulk file records to the appropriate Terminal files records, the Bulk and Terminal file data are matched on CusWC and TERM_ID and merged into an interim database with the following structure:

CUSWC - Customer wire center
PHONE - Customer phone number
CUSTERM4 - Terminal ID from Bulk file
TERM_ID - Terminal ID from Terminal file
CLASS - Class of service
CUSADD - Customer address from Bulk file
TERMADD - Terminal address from terminal file
CUSUNIT - Used from multi-family or multi office units
TYPE - Terminal type
SIZE - Terminal size/capacity
COUNT_CAID_PAIR/RANGE - Cable count of terminal
CUSCITY - City name from Bulk file
CUSSTATE - State name from Bulk file

CUSZIP - Zip Code from Bulk file

IPID – (Individual Plant ID): IPID # of terminal (incomplete data)

This interim database forms the 'bridge' between the Terminal and Bulk file data. The primary purpose of this table is the association of the appropriate Zip Code in the Bulk file to the terminal address, allowing the terminal location to be accurately geo-coded. In addition, this interim table forms the base table for further preparation of the data for geocoding. The records in the Bulk table not included or matched in the Terminal table are not discarded, but are retained in the Bulk table to form the total switched demand quantities for the model.

3.3.5 Preparation of table for improved Geo-coding results

The data in the interim table could be geocoded at this time in the process and would probably obtain reasonable results. However, several techniques have been implemented by VzLoop developers to increase the number of valid, geo-codeable addresses in the database.

Instances occur where a terminal has no address information. To alleviate this situation, the interim table is sorted on CusWC, TermAdd, and TermID, and terminals with no address are given the service address of the customer being served by the terminal. In cases where the terminal serves more than one address, the service address with the lowest numerical address is chosen and used as the terminal address.

Other instances occur where the terminal has an address but the service location(s) served by the terminal do not contain a zip code. In this case, the interim table is searched for a service address with the same street name as the terminal location in question, and the zip code of that location is used for the terminal location. If no matching street name is found, the predominant Zip Code for that area is used.

3.3.6 Geo-coding discussion and preparation of interim table for geo-coding

After all modifications to the terminal data are completed, the data not essential for geo-coding is stripped from the interim files. The structure of the final file sent for geo-coding is as follows:

WIRE CENTER
TERMINID
ADDRESS
CITY
STATE
ZIP CODE

To the above structure, the geo-coding process adds the fields latitude, longitude, and georesult to the file. Georesult is the degree of accuracy obtained by the geo-coding

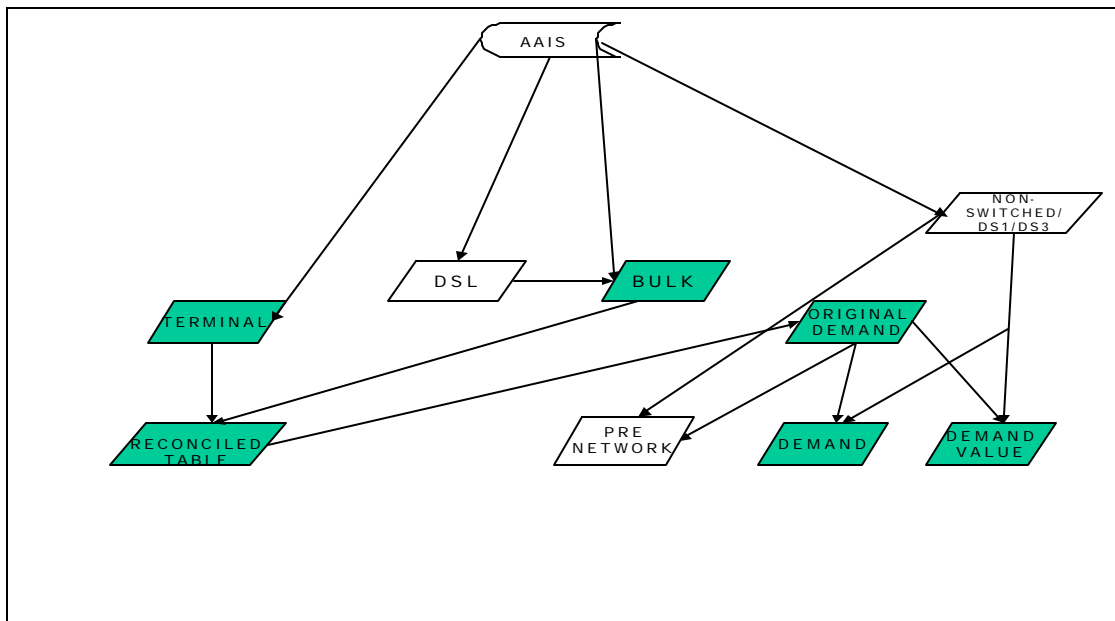
process. There are six levels of geo-coding accuracy, listed in order of most to least accurate:

- Exact : To the doorstep
- ShapePt: On the shape of the road
- Zip+4: Center of the city block
- Zip+2: Within 2 square miles
- ZipCode: Centroid of Zip Code Area
- NoMatch: No matching information

Terminals having an accuracy level of exact, shapePt, or Zip+4 are acceptable for terminal location. Terminals with an accuracy level of Zip+2 or ZipCode are acceptable only as a last resort, and nomatch is not acceptable. Actions taken throughout the pre-processing procedure strive to improve the accuracy of the locations of terminals with an accuracy level of zip+2, zipcode, and nomatch.

3.4 Creation of Demand and Demand_value tables

The data in VzLoop's Demand and Demand-value tables originate from the AAIS Bulk file. The Demand and Demand-value tables are constructed directly from the file resulting from the Bulk file and Terminal reconciliation, (the OriginalDemand Table). Non-switched and DS1 demand is added later in the process. The Demand and Demand-value tables are VzLoop system tables in that, unlike the tables discussed up to this point, cannot undergo a table structure change without extensive VzLoop program modification. Therefore, any type of table changes dealing with structure is not allowed for these two tables.



Demand/Demand Value table development

3.4.1 Demand table

The permanent structure of the Demand table is:

Jurisdiction
CLLI
Term
LUID
TermUBA
Totalres
Totalbus
Vintage
TotalDS1

3.4.2 Demand table description

The Demand table is a listing of customer service addresses with their associated demand. VzLoop uses the demand table to place customer service demand at the appropriate terminal location in the network file. It is also used to determine the quantity and size of service drops and NIDs required in the network, as each record in the demand table is considered for drop and NID placement.

The records in the demand table are unique on jurisdiction, CLLI, Term, and LUID (living unit ID) and reflect the total residence, business, and DS1 lines at a specific service address. The jurisdiction, CLLI, and term fields are the key fields VzLoop uses to place the demand of the LUID into the network file. The term to LUID relationship in the demand table can be 1 term to 1 LUID or 1 term to many LUIDs, the latter reflecting cases where terminals serve more than one service address.

The TermUBA field in the Demand table is used to determine the construction type of the drop.

The Totalres and Totalbus fields are the summation of all lines at that LUID that have been classified as residence or business lines, as explained in the following paragraphs.

The TotalDS1 field is the summation of all DS1 facilities at that LUID.

Development of demand table data

Before the Demand table is populated, the detail service types need to be identified. Detail service type identification needs to be done on a state basis as slight differences exist in service types and service type naming conventions among the states. A query named 'Distinct Class of Service' has been developed for this purpose and is run against the OriginalDemand Table. After the query is run and all service types for a particular state are identified, any manual additions or deletions of detail service types to the table will take place.

The result of this effort is the Detail service type list in the table below. Note that this is a sample conversion table.

Sample Class of Service Designation Table

Detail service type	Service type	Service class
B1	BUS	B
SFBUS	BUS	B
CTX	CNTRX	B
LHCTX	CNTRX	B
LHSS	CNTRX	B
COIN	COIN	B
CKT	NSW_PL	B
ADVSVC	OTH	B
NONE	OTH	B
TEST	OTH	B
LD	PBX	B
LH	PBX	B
SPL	SW_PL	B
IWTS	WATS	B
OWTS	WATS	B
INRNET	OTH	B
MPTY	RES	R
PTY2	RES	R
R1	RES	R
SFRES	RES	R
DSL	BUS	B
RESDSL	RES	R

Following the development of the table above, extensive data preparation still remains to convert data from the OriginalDemand Table to the Demand table. Data resides in the OriginalDemand Table on a telephone number basis and contains, among other items, the detail service type of that telephone number. Take, for example, a service address in the OriginalDemand table (LUID1), served from terminal TERM1, with two single party residence lines. This service location requires two records in the OriginalDemand Table (two telephone numbers), each having a detail service type of R1. A residence next door (LUID2), served from TERM2 may also have two lines, one R1 and one B1 detail service type. As with LUID1, LUID2 also requires two records in the OriginalDemand table. Conversion of this data to the demand table format requires two actions take place. First, using the above table, the service type must be converted to its appropriate service type and then to its appropriate service class (R1-RES-R, B1-BUS-B). Second, the service class must be totaled on a Jurisdiction, CLLI, and LUID (address) basis. The result of this procedure is to produce two records in the Demand table as follows:

Version_ID Jurisdiction CLLI Term LUID TermUBA Totalres TotalbusVintage

```
## ### ##### TERM1 LUID1 2 0 ##  
## ### ##### TERM2 LUID2 1 1 ##
```

The Demand Table is now complete for switched lines.

3.4.3 Integration of non-switched/DS1 lines into the Demand table

The demand table discussed up to this point contains switched and DSL lines only. After the Demand table is completed for those lines, non-switched and DS1 lines are appended to the table to more accurately reflect the entire gamut of lines served by the outside plant network. As mentioned earlier in the document, the non-switched lines are obtained from the AAIS-Core group upon request and the DS1 information is obtained from the AAIS-DS group. Non-switched line demand is added to the Totalbus quantity of the appropriate terminal through a terminal matching process with the circuit table. DS1 demand is added to the TotalDS1 quantity of the appropriate terminal by first geocoding the DS1 file and then determining the nearest (geographically speaking) terminal to the geocoded DS1 location.

3.4.4 Demand_value table

The permanent structure of the Demand_value table is as follows:

```
Item_key  
Geography_key  
Value
```

The Item_key field describes the service type (examples shown in table above), the geography key contains CLLI code, and value is the number of lines for that service type in that CLLI code.

3.4.5 Demand_value table description

The demand_value table is used by VzLoop to provide line totals by service type per wire center. The table also provides VzLoop with a total of residential and business lines per wire center. The line totals in the demand_value table must agree with the line totals in the demand table.

Development of demand_value table data

Shown below is a sample of the demand_value table. The items under the item_key are fixed, in that they are a listing of service types, repeated for each wire center, and cannot be changed. As seen from the sample Demand Value table below, service types start with TOTALRES and continue to NWKG, then repeat for each geography key.

Sample Demand Value table

ITEM_KEY	GEOGRAPHY_KEY	VALUE
TOTALRES	CLLI CODE1	1271
TOTALBUS	CLLI CODE1	174
RES	CLLI CODE1	1271
BUS	CLLI CODE1	92
WATS	CLLI CODE1	0
PBX	CLLI CODE1	13
CNTRX	CLLI CODE1	4
COIN	CLLI CODE1	3
BRI	CLLI CODE1	0
RESDSL	CLLI CODE1	0
DSL	CLLI CODE1	0
DDS	CLLI CODE1	0
SW56	CLLI CODE1	0
NSW_PL	CLLI CODE1	2
SW_PL	CLLI CODE1	0
DS1	CLLI CODE1	0
PRI	CLLI CODE1	0
OTH	CLLI CODE1	60
NWKG	CLLI CODE1	0
TOTALRES	CLLI CODE2* Begin next Clli	1064
TOTALBUS	CLLI CODE2	234
RES	CLLI CODE2	1064
BUS	CLLI CODE2	108

The data for the demand_value table is developed against the ORIGDemand Table, with the Class of service as rows, the wire center or CLLI as column headings, and phone numbers as count. The class of services are summed into the appropriate 'totalres' or 'totalbus' lines. At this point in time, RES and RESDSL map to totalres and the remaining services except DS1, map to totalbus.

3.5 Terminal_snap and Pre_network tables

The terminal_snap and the pre_network tables originate from AAIS. Programs (to be discussed later) run with the terminal_snap table provide updates to the termuba, rx, ry, and family fields in the pre-network table. Table structures are as follows:

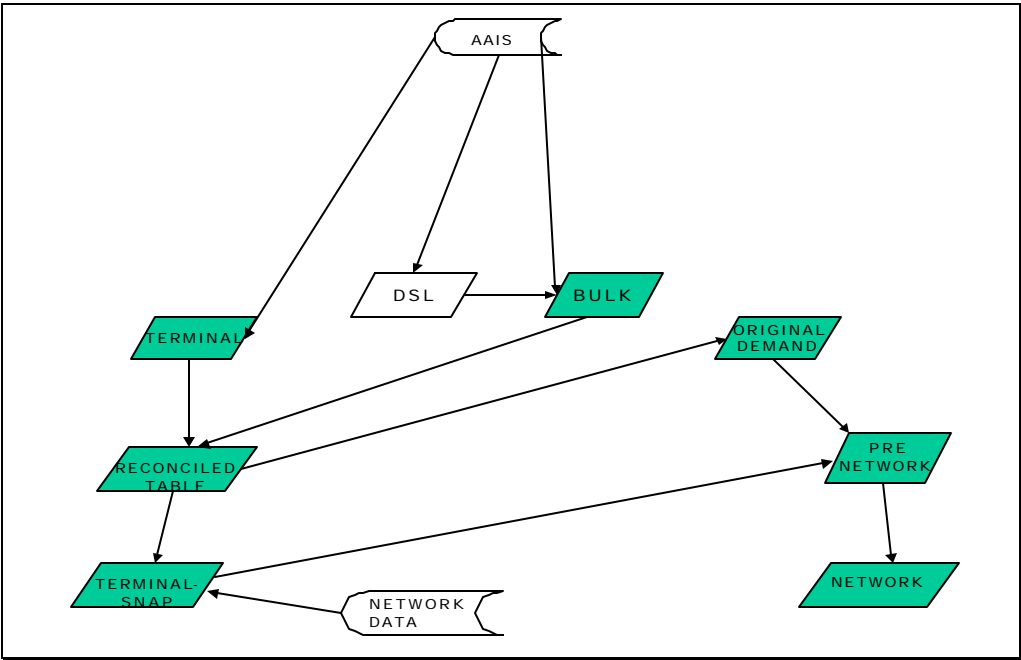
TERMINAL_SNAP TABLE

CLLI
 WC
 TERM
 FAC_SIZE

PRE_NETWORK TABLE

CLLI
 TERM
 RTE
 X

COUNT	Y
LONGITUDE	RX
LATITUDE	RY
RELX	TOTALRES
RELY	TOTALBUS
GEORESULT	TERM_TYPE
NEWX	TERMUBA
NEWY	CPUS
TERMUBA	RCP
FAMILY	FAMILY
SUBFAMILY	



Pre-Network and Terminal-Snap table development

3.5.1 Development of Terminal_snap table

The terminal_snap table is used by the loop pre-process to house geocoded and network-generated terminal locations and network-generated terminal grouping and construction type data. This terminal_snap data is then transferred to the pre-network table, which in turn is used to generate the VzLoop network table. The terminal_snap table is developed initially from the post-geocoded terminal table and additional data is added from the network processes, to be discussed later. Specifically, data fields WC, term, fac_size, count, longitude, latitude, relx, rely, and georesult data are populated directly from a download of the post-geocoded terminal file. The remaining fields in the terminal_snap table are populated by the network process. The only data manipulations required for the initial development of the terminal_snap table from the post-geocoded terminal file is a

matching of the terminal_snap WC field to a WC-CLLI conversion table to populate the CLLI field and modifications to several of the georesult types. To comport to the termsnap.exe program, which is discussed later, georesults of 'ZIP+2' are changed to 'NOMATCH' and georesults of ZipCode' are changed to 'NoMaTcH'. After these changes are made, the base terminal_snap table is ready to be processed.

3.5.2 Development of Pre_network table

The pre_network table forms the basis for the VzLoop network table in that, in its final form, it contains all the terminals, cross connects, pair gains, and associated information required to develop the facility routing in the network table. The initial population of the pre_network table requires that terminal data be placed in the following fields: CLLI, term, totalres, and totalbus. This data is acquired through a direct download of selected fields in the OriginalDemand table. As in the terminal_snap table, remaining data fields in the pre_network table are populated through the use of a program, to be discussed later. Following the population of the CLLI, term, totalres, and totalbus fields, the pre_network table is ready to be processed. At this point, all activities that deal exclusively with AAIS data are complete.

3.6 Summary of AAIS - Derived Tables

AAIS is the source of all terminal, DSL, customer, non-switched, and DS1 circuit information in the VzLoop pre-processing. In addition, through the use of gecoding, AAIS forms the basis for the locations of terminals in the loop pre-processing. Tables populated directly by AAIS data are the Terminal, Bulk, DSL, non-switched, and DS1 tables, which in turn are used to populate the Demand, Demand Value, Terminal_Snap and Pre_Network tables.

The Demand and Demand Value tables are used directly by VzLoop and their structures cannot be modified without a major programming change in VzLoop. The Terminal_Snap and Pre_Network tables are modified further in the pre-process by network data (see following sections), and provide the data required for the development of the VzLoop Network table.

4.0 Network Input into the Pre-process

The focus of the document now shifts to the contribution of network and planning data to the loop pre-process. Up to this point, the detailed portion of this document dealt exclusively with data gathered from the AAIS. The following sections deal with the gathering of existing network and planning data, reasoning associated with the application of the network data to AAIS data, and a step-by-step explanation of the process.

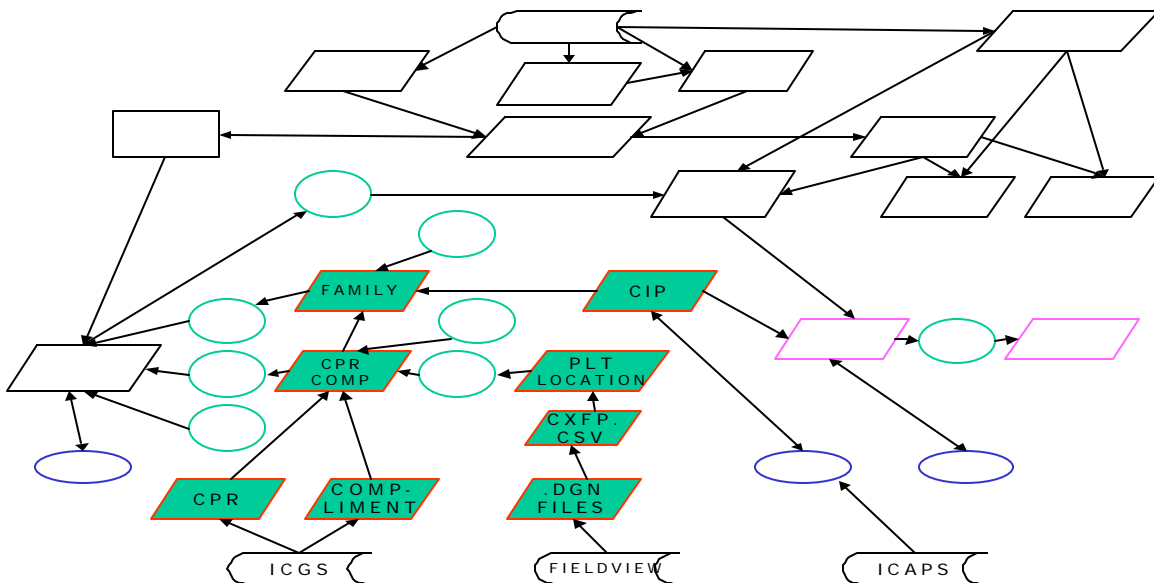
4.1 Table Development from ICGS/Fieldview and ICAPS

The pre-network table developed up to this point in the AAIS portion of the pre-process could be used, with logical adjustments, to create a Network table for use by the VzLoop Loop Module.

The purpose of the ICGS/FIELDVIEW/ICAPS, ie, the ‘network’ portion of the VzLoop pre-process is to vastly improve the quality of the data in the Network table by improving the accuracy of terminal locations and providing existing feeder routes, terminal groupings to existing DLCs and cross-connect boxes, and construction types. What follows is a description of the processes used to accomplish these data improvements,

4.1.1 File Development

Shown below is a diagram depicting ICGS, Fieldview, and ICAPS-derived tables developed for use in the pre-process. This portion of the document looks at the acquisition and development of each of the ICGS/ICAPS tables.



ICGS, FieldView, ICAPS – derived tables

4.1.2 ICGS, FieldView, ICAPS

Following is a description of the ICGS, Fieldview, and ICAPS systems and actions and queries used to access and retrieve the appropriate data. from these systems.

4.1.3 ICGS

As mentioned in the introduction, ICGS is a Verizon legacy system. It provides OSP facility data, including cable type, cable length, cable/component size, account code, etc. for use on a daily basis by Engineering and Accounting personnel. Data is extracted from

the ICGS CPR and COMPLIMENT files for use in the VzLoop preprocess in the CPRCOMP table.

4.1.4 ICGS CPR file

As its name implies, the ICGS CPR(Continuing Property Records) file houses data required to account for outside plant equipment. The CPR file contains many fields required for work order tracking and other activities not required by VzLoop, therefore, not all information is extracted from this file. The following fields are extracted:

CPR file structure

Jur Wire center IPID Category Account_cd Pairsize

A typical record in the CPR file might be:

WA 4520 553 CABLE 242110 200

where:

WA = Washington jurisdiction

4520 = ICGS designation for ORA LOMA wire center

553 = IPID number*

CABLE = Identifies a copper cable

242110 = Account code for aerial copper cable

200 = Cable size, in pairs

* IPID (Individual Plant ID) numbers identify the smallest unit of plant in the ICGS system.

4.1.5 ICGS Compliment file

The ICGS Compliment file houses data that identifies cable count, or cable complements, for items of plant where this type of information applies. It, like the CPR file, contains data not required by VzLoop, so not all information is extracted. The following is extracted:

COMPLEMENT file structure

Jur Wire center IPID source_ipid cable_id low_pair high_pair

The record in the COMPLEMENT file for the same IPID shown above in the CPR file might be:

WA 4520 553 545 C2 201 400

Information in the Complement file not identified in the CPR file example are:

545 = Source IPID = The IPID connected to IPID 553 on the central office side of IPID 553.

C2 = Cable_ID. The identifying designation of a cable count or complement.

201 = The lowest numerical pair identifier in the cable count, or compliment.

400 = The highest numerical pair identifier in the cable count, or compliment.

In the above example, the complete cable complement for IPID 553 is C2, 201-400, which indicates a 200 pair cable. This is confirmed by the pair-size indicator of '200' in the CPR file.

4.1.6 FIELDVIEW

The FIELDVIEW system is actually a graphical subsystem of ICGS. FIELDVIEW contains graphical files that display the outside plant network on a computer monitor in much the same way as paper plant records once did. As in the ICGS files, the FIELDVIEW files, called .dgn files, contain much more data than what is needed by the VzLoop pre-process. The VzLoop extraction process captures the pertinent data from the .dgn files and stores that data in .jnl, or journal files. The journal files are then consolidated into the **PltLocation** table for use by the VzLoop preprocess.

4.1.7 ICAPS

ICAPS (Integrated Customer Access Planning System) is an OSP planning tool used to monitor utilization of feeder plant. In order to provide accurate and timely feeder plant monitoring using the ICAPS, Network Planners build an electronic version of the feeder network with a series of control points along each feeder route. This network layout, or control point schematic, contains data required for the planner to effectively monitor the feeder network. Data segments between control points contain items such as cable type, size and quantities, lengths of segments, construction type, distance and direction from the wire center, etc. In conjunction with ICAPS, a separate database called NETCAPS contains DLC and cross-connect quantities and locations. The ICAPS/NETCAPS provides a very complete and realistic snapshot of the existing feeder network.

VzLoop pre-processing takes advantage of the prior and present efforts of the Planning department and uses items in the ICAPS schematic and NETCAPS database as the basis of its feeder network. Items gleaned from the ICAPS schematic include feeder cable locations, connectivity, and cable construction types. DLC and cross connect locations, quantities, and designations are extracted from the NETCAPS database through an on-line query called NETCAP ONLINE. After processing, data from the ICAPS and NETCAP ONLINE are populated through the WANET, WADLC, and WAXBOX files into the **CIP** table for use in the VzLoop preprocess.

4.1.8 ICAPS WANET file structure

Data extracted from the ICAPS schematic is stored in a file called WANET. Data fields used in the VzLoop preprocess are as follows:

CLLI – Standard eight digit wire center identifier

CPUS – Control Point Under Study. Data in the record is associated with the CPUS

RCP – Related Control Point – The control point connected to the office side of the CPUS.

CPUTMX – The UTM x coordinate of the CPUS *

CPUTMY – The UTM y coordinate of the CPUS

CONSTYPE – The construction type(s) of the cable(s) in the segment (between the CPUS and the RCP)

*UTM - Universal Transverse Mercator – A grid-based mapping system. A form of this mapping system is used by ICGS to determine plant locations. In turn, ICAPS uses the UTM coordinates from the ICGS to locate control points.

4.1.9 NETCAPONLINE WADLC and WAXBOX file structure

On-screen querying features in NETCAPONLINE allow for a download of applicable DLC and cross-connect information into tables called WADLC and WAXBOX, respectively. The downloaded files contain many fields not required by VzLoop. The retained data fields for the WADLC table are:

CLLI
CPUS
IPID

The CPUS field is used to key to the WANET table to obtain location coordinates. The IPID field is used to key to the CPRCOMP table to obtain the cable count of the DLC, in order to group the appropriate terminals to the DLC.

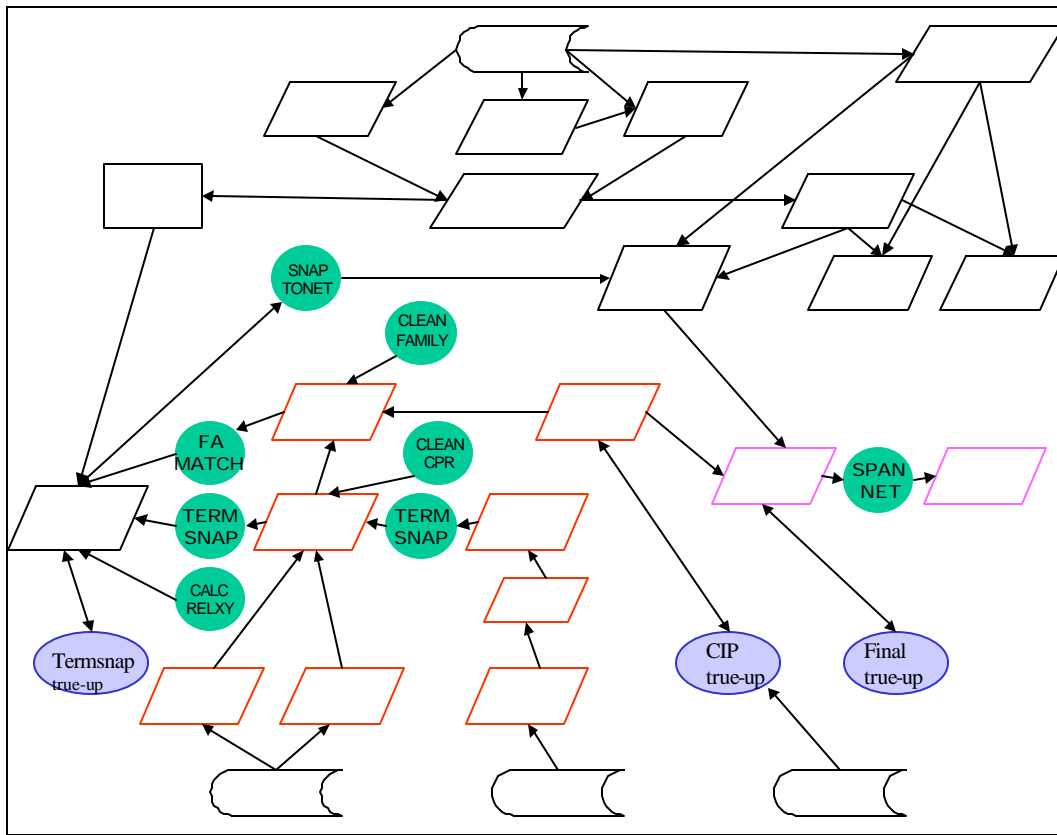
The retained fields for the WAXBOX table are:

CLLI
CPUS
IPID
XBOXID

The same reasoning applies for the WAXBOX table. In addition, the XBOXID field provides an additional method of identifying the cross-connect box out-count cable id, in order to group the appropriate terminals to the cross connect box.

5.0 Programs and Processes

Several programs and processes are required to perform the activities required to process the data gathered from the AAIS, ICGS, FIELDVIEW and ICAPS systems. Programs are grouped as major or minor according their complexity and logic. Processes are defined as manual checks and modifications to the established programs and tables.



Pre—processing programs and processes. Programs shown as circles. Major programs bolded. Processes shown as ovals.

5.1 Processes

1. Terminalsnap true_up
2. CIP true_up
3. Final true_up

5.1.1 Terminalsnap True Up

The terminalsnap trueup process consists of two sets of adjustments. One concerns the selection of the geocoded terminal locations versus the terminal locations derived from the cable plant records and is run after the termsnap.exe and before the Famatch.exe. The other involves the possible redirection of terminals to more logical families, and is run after the Famatch.exe.

The initial terminalsnap trueup provides for a selection of geocoded terminal locations based on the geocoded accuracy level and a predetermined proximity. If a terminal has a geosresult of Exact, ShapePt, or ZIP+4 and the geocoded location is within a predetermined distance from the cable record derived location, the geocoded location is

used. It also provides for the use of the geocoded location if the cable derived location is null, the georesult is Exact, ShapePt, or ZIP+4, and the geocoded location is inside the wire center boundaries. The wire center boundary is defined as an area bounded by the minx, maxx, miny, and maxy coordinates of the cable plant records of the wire center.

The second terminalsnap true-up provides the user with steps to redirect terminals to more logical families or locations, thereby reducing the possibility of exaggerated distribution routes.

The first step is to replace all null family fields with a family field of 'CO'. This happens very infrequently and results in a terminal being routed to the nearest control point, thereby bypassing any cross connect or DLC it may have been associated with. This in turn results in a slight undersizing of cross connects and DLCs.

The second step involves records with geocoded and cable-derived locations of 0,0. In this case, the terminal is placed at the cross connect of DLC of its non-CO family where possible.

If, after step 2, some terminals with a non-CO family remain at 0,0, their family is changed to CO. This prevents any erroneous placement of distribution cable to the terminal.

5.1.2 CIP Trueup

The CIP trueup can be run anytime before the CIP table is uploaded into the Pre-network table. This process involves testing the control point schematic for continuity as well as for limiting all related control points to no more than four links to various control points.

It also removes any duplicate control point/related control point records from the CIP table.

5.1.3 Final Trueup

The final trueup is done immediately preceding the SPANNET.exe. It insures that all related control points in the pre-network table representing the central office are listed as '000'. It also deletes all un-located serving terminal records and records with no demand at this time, thereby eliminating terminals with no associated plant.

5.2 Minor Programs

1. CleanCPR.exe
2. Cleanfamily.exe
3. Calcrelxy.exe
4. SnaptoNet.qry

5.2.1 CleanCPR.exe

CleanCPR.exe is the first program run in the preprocess. It is run before the Termsnap.exe program.

Cable Ids are designated differently in the CPRCOMP and the Terminalsnap tables. Numeric cable counts in the CPRCOMP table are preceded with a 'C'. In the Terminalsnap table, numeric cable counts are not preceded with a 'C'. For example, Cable 1 is shown as 'C1' in the CPRCOMP table and as '1' in the Terminalsnap table. Nonnumeric cable Ids, such as AA, BE, etc. are identical in both files. In order for the numeric Cable Ids to have a possibility of matching in the Termsnap program, the leading 'C' needs to be removed from the Cable ID field of the CPRCOMP table when the 'C' is followed by a numeric. That is the function of the CleanCPR.exe program.

5.2.2 CalcRelxy.exe

The function of the CalcRelxy program is to calculate relative x,y positions for the terminals in the Terminalsnap table prior to running the Termsnap program. Terminal locations must be converted to relative x,y positions to provide a common basis for locations of network items added later in the process. It does this by comparing the latitude and longitude of the terminals from the geocoding process to the latitude and longitude of the central office, using the central office as the geographic center of the wire center, or $x=0,y=0$.

5.2.3 CleanFamily.exe

The CleanFamily.exe is run after the Termsnap.exe and before the Family.exe. It performs the same function to the Family table as the CleanCPR.exe does to the CPRCOMP table.

5.2.4 SnaptoNet.qry

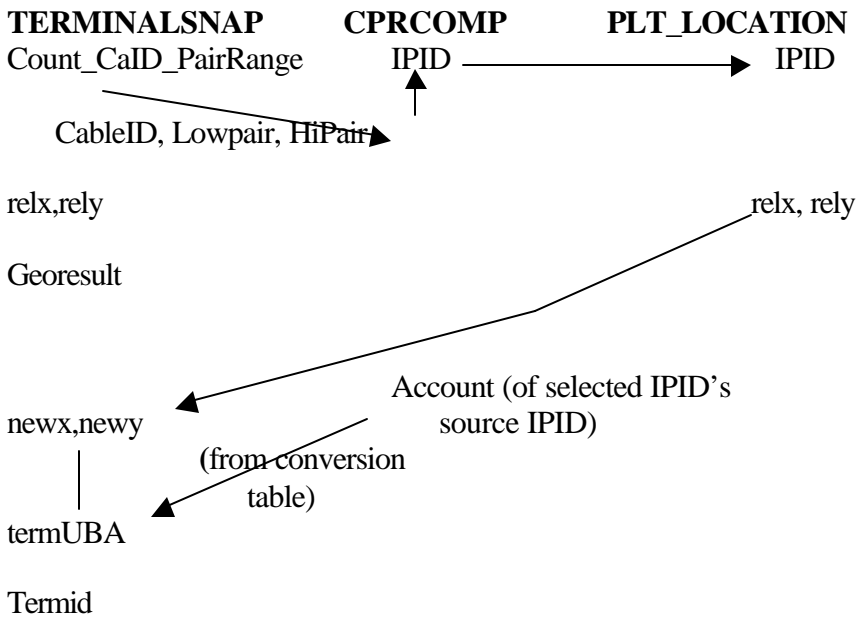
The SnaptoNet.qry is a .SQL query that is run after the Termsnap.exe and FAMATCH.exe. It is used to transfer location, family and TermUBA data from the terminals in the Terminalsnap table to their corresponding terminals in the Pre-network table.

5.3 Major Programs

1. Termsnap.exe
2. Famatch.exe
3. Spannet.exe

5.3.1 Termsnap.exe

The termsnap.exe program provides for proper terminal locations relative to actual cable routes and for proper population of the TermUBA field in the prenetwork file. The process involves three tables (plt_location, terminalsnap, and cprcomp) . Termsnap.exe is performed for all records (terminals) in the terminalsnap file. The process consists of various methods of matching the cable count of a terminal in the terminalsnap file with the appropriate cable IPID in the cprcomp table having the proper cable count, as detailed in the following paragraph. Once the appropriate match is made, the cable IPID's relative x,y position in the network (from the plt_location file) and the cable IPID's source IPID's construction type (from the CPRCOMP file) are populated in the terminalsnap file for that terminal.



Process flow of termsnap.exe

5.3.1.1 Comparison of Terminal Count to Cable Count

Described several paragraphs below is the process used for identifying and acting on various matching scenarios that exist between the terminalsnap and cprcomp tables. Before matching can take place, however, data in the data fields relating to cable and pair identification in the terminalsnap and cprcomp files must be correctly identified. The details involved in this identification are as follows:

Terminalsnap file

Terminal count is contained in the terminalsnap file in the field labeled 'Count_CaID_PairRange'

Below is an extract of a portion of several terminalsnap file records to show several variations of the Count_CaID_PairRange field.

Size	Count_CaID_PairRange
25	30-151/175
200	221-1901
300	217-251/300 + 25-901/1000
200	UR-1/122 + UR-127/140 + UR-145 + UR-150/200
100	14-1/6 + 94D519-1:1/48 + BY1-126/144
50	94D519-2:49/96

Records 1-4 (records with no ':')

For records with no colon, Cable ID consists of the digits immediately followed by a '-'. Pair range as discussed below are the digits immediately following the '-'. This can be a single pair, as shown in the second record in the example, or a range of pairs, as shown by the other records. For example, record 1 reflects pairs 151 to 175 in cable 30. Most records will contain one cable ID and one pair range, as shown in the first two records. However, some records will contain one distinct cable ID with multiple cable ranges (separated by a '+', as shown in the fourth record) and a few will contain multiple cable IDs and multiple pair ranges (separated by a '+', as shown in the third record.)

Records 5-6 (records with ':')

A very small percentage of records contain a colon in the Count_CaID_PairRange field. They must be treated differently than records without a colon.

As in records without colons, records with colons can contain various mixes of Cable IDs and pair ranges. Two types are shown in records 5 and 6. Record 5 shows multiple cable IDs and multiple pair ranges (separated by a '+'). Record 6 contains one distinct cable ID with one cable range. Regardless, if the record contains a colon, the following is done: If the group string (digits between the '+') contains a ':', Cable ID is the digit string immediately followed by a ':'. If the group string does not contain a ':', Cable ID is the digit string immediately followed by a '-', as in the preceding paragraph. For example, in record 5, Cable IDs are '14', '94D519-1', and 'BY1'. The Cable ID in record 6 is '94D519-2'

Pair ranges for the group strings with a colon are the digits immediately following the ':'. Pair ranges for the group strings without a colon are the digits immediately following the '-'. For record 5, pair ranges are '1/6', '1/48', and '126/144'. The pair range for record 6 is '49/96'.

CPRCOMP Table

Cable count is contained in the CPRCOMP file in the fields labeled 'Cable_ID, LowPair, and HiPair'

Below is an extract of a portion of several CPRCOMP records to show several variations of the 'Cable_ID, LowPair, and HiPair' fields.

SourceIPID	Cable ID	LowPair	HiPair	Acct	Size
	333C33	671	671	242110	900
	370XK	201	250	242110	100
	28200LW	76	100	242310	25
	49258C230E	126	200	242110	100
	-10XK	701	725	242110	25
	369XK	1	100	242110	100

The Cable ID field in the CPRCOMP file is analogous to the Cable ID portion of the Count_CaID_PairRange field in the Terminalsnap file, with one exception. If the Cable ID field begins with a 'C' and is followed by a numeric digit, the C must be stripped from the record. For example, Cable ID '223' in the terminalsnap file will be shown as 'C223' in the CPRCOMP file and the 'C' will need to be stripped. However, cable ID 'CA' in the terminalsnap file will also be shown as 'CA' in the CPRCOMP file and no stripping is required. The digit stripping process is performed in the CleanCPR program, as explained earlier.

The LowPair field is analogous to the first value of the pair range in the terminal field (the value immediately preceding the '/' in the terminalsnap file) and the HiPair field is analogous to the second value of the pair range (the value following the '/'). Records reflecting one cable pair are shown as record one in the sample CPRCOMP file, with 'LowPair' equal to 'HiPair'.

After the data field identifications are complete, matching between the terminalsnap and CPRCOMP files are ready to begin.

5.3.1.2 Scenarios Relating the Terminal Count to the Cable Count

The following cases describe actions taken to accommodate the various scenarios encountered in matching records in the terminalsnap file to cable IPIDs in the CPRCOMP table.

Cases 1-2 deal with terminals that have reliable lat/long coordinates (georesult=<>'NoMatch' and <> 'NOMATCH' and <> 'NoMaTcH').

Cases 3-4 deal with terminals that have lat/long coordinates, but apply only to specific GEORESULT levels.

Case 5 deals with all terminals where georesult='NoMatch' or 'NOMATCH' 'NoMaTcH'.

CASE 1: Terminal count contains one cable ID and one pair range (Records 1,2 and 6 of the sample terminalsnap file) and georesult<>'NoMatch' and <> 'NOMATCH' and <> 'NoMaTcH'

- a. Using the CLLI code, cable ID and pair range information from the terminalsnap file, identify records in the CPRCOMP file with matching CLLI code and cable Ids, and pair ranges that match or include the pair range from the terminalsnap file. Retain the IPID numbers of the qualifying IPIDs.
- b. Using the retained IPID numbers, search the Pltlocation file for these IPIDs and select the IPID in the Pltlocation file with the relx-rely coordinate **nearest** to the relx-rely coordinate of the terminal.
- c. Place the relx-rely coordinates of the selected IPID from the Pltlocation file into the Newx and Newy fields of the specified terminal record in the Terminalsnap file.
- d. Replace the TermUBA field in the Terminalsnap file with the appropriate U, B, or A based on the account code of the selected IPID's **source IPID** from the CPRCOMP table. An additional table will be provided to provide an 'account to UBA' conversion.

CASE 2: Terminal count contains one unique cable ID and more than one pair range (Record 4 of the sample terminalsnap file) and georesult<>'NoMatch' and <> 'NOMATCH' and <> 'NoMaTcH'

- a. Using the CLLI code, cable ID and multiple pair range information from the terminalsnap file, identify records in the CPRCOMP file with matching CLLI code and cable Ids, and pair ranges that include the entire pair range from the Terminalsnap file record. For example, the fourth record of the sample Terminalsnap file have a count of UR-1/122 + UR-127/140 + UR-145 + UR-150/200. Qualifying IPIDs from the CPRCOMP file must include the entire cable range, therefore must include Cable ID UR with a low pair of '1' and a minimum 'HiPair' of '200'. Retain the IPID numbers of the qualifying IPIDs.
- b. Using the retained IPID numbers, search the Pltlocation file for these IPIDs and select the IPID in the Pltlocation file with the relx-rely coordinate **nearest** to the relx-rely coordinate of the terminal.
- c. Place the relx-rely coordinates of the selected IPID from the Pltlocation file into the newx and newy fields of the specified terminal record in the Terminalsnap file.
- d. Replace the TermUBA field in the Terminalsnap file with the appropriate U, B, or A based on the account code of the selected IPID's **source IPID** from the CPRCOMP table.

CASE 3: Terminal count contains more than one unique cable ID and GEORESULT is 'ShapePt' or 'ZIP+4' (Record 3 and 5 of the sample terminalsnap file, assuming GEORESULT is ShapePt' or 'ZIP+4')

This scenario is rare and usually indicates that more than one cable IPID is serving the terminal. However, because the geocode in this scenario did not return an exact location, the following procedure will be used to position the terminal and determine the TermUBA.

- a. Using the CLLI code, multiple cable ID and multiple pair range information from the terminalsnap file, determine which cable ID is dominant, i.e., has the largest pair range. For example, in the terminalsnap file sample, Cable ID '25' is the dominant cable ID because its pair range is 100 (901-1000), while Cable ID '217' has a pair range of 50. Once the dominant ID is determined, search for qualifying IPIDs from the CPRCOMP file that include the entire pair range of the dominant cable ID. Retain the IPID numbers of qualifying IPIDs.
- b. Using the retained IPID numbers, search the Pltlocation file for these IPIDs and select the IPID in the Pltlocation file with the relx-rely coordinate **nearest** to the relx-rely coordinate of the terminal.
- c. Place the relx-rely coordinates of the selected IPID from the Pltlocation file into the newx and newy fields of the specified terminal record in the Terminalsnap file.
- d. Replace the TermUBA field in the Terminalsnap file with the appropriate U, B, or A based on the account code of the selected IPID's **source IPID** from the CPRCOMP table.

CASE 4: Terminal count contains more than one unique cable ID and GEORESULT is 'Exact'. (Record 3 and 5 of the sample terminalsnap file, assuming GEORESULT is 'Exact')

As in Case 3, this scenario indicates a multi-cable feed to the terminal. In this scenario, however, since the geocode is exact, no action will be taken to reposition the terminal. In addition, TermUBA for this terminal will be derived from the TermUBA of the nearest terminal.

CASE 5: Terminal was unable to be geocoded reliably (GEORESULT is 'no match', NOMATCH, or NoMaTcH).

Using the CLLI code, cable ID and pair range or multiple pair range information from the terminalsnap file, identify records in the CPRCOMP file with matching CLLI code and cable ID and pair ranges that include the entire pair range from the terminalsnap file record.

- a. From this set of IPIDs, select the IPID(s) with the **smallest cable size** (CPRCOMP 'Pairsize'). If several IPIDs have the same size, select the first IPID in the list with the smallest cable size. The selection of the first qualified IPID should prevent criticism concerning 'arbitrary' placement of terminals. Retain the IPID number.
- b. Using the retained IPID number search the Pltlocation file for this IPID.

- c. Place the relx-rely coordinates of the selected IPID from the Pltlocation file into the newx and newy fields of the specified terminal record in the Terminalsnap file.
- d. Replace the TermUBA field in the Terminalsnap file with the appropriate U, B, or A based on the account code of the selected IPID's **source IPID** from the CPRCOMP table.

5.3.2 Famatch.exe

The FAMATCH program is used to assign appropriate family designations to terminals, thereby enabling a logical grouping of terminals. Before the FAMATCH.exe can be explained, however, an understanding of network families and the processes required before running FAMATCH is needed.

Understanding Network Families and Sub-families

In today's local telephone network, outside plant facilities are assigned, managed, and engineered through the use of the cable count of the facility. Cable counts include a cable or system number and a pair or channel range. Except for the very smallest wire centers, cable counts are generally confined to pre-determined areas, with as little cable count overlap as practical. In addition, , the area served by a cross connect box should change very little once established, and the area should be served exclusively by that box.

By general definition, a network family consists of facilities with a common and unique cable count or system number. For example, all customers whose terminal has a central office cable count of Cable 1 can be considered as part of the Cable 1 network family and will share the central office as a common node (a CO family). All customers working on a terminal with cross-connect count AA are considered as part of the Cable AA network family and will terminate at the AA cross connect box, where their demands will be consolidated and routed to the central office. Likewise, all customers whose terminal contains DLC count C103 will terminate at the DLC C103 location and their demands consolidated onto a fiber facility to the central office from that DLC.

As is often the case, exceptions to the rule exist in the existing network. Some unique cable counts are not routed all the way back to the office, a cross connect box, or DLC. For example, assume the following: Cable count AA, 1-100 serves as the in-count to a large building terminal. For local network management purposes, the terminal out-count was designated BLDG 1-100. In this case, cable BLDG, even though it is a unique cable ID, would not warrant an exclusive route back to the cross-connect box. In fact, it would not even route back to the cross-connect box as Cable BLDG. Its demand would be absorbed as part of the AA network family and routed to the cross-connect as part of cable AA. Cable BLDG would be defined as a network sub-family and consolidated into network family AA.

The VzLoop pre-processing program utilizes facilities' cable counts and locations to group cable plant into logical network families. The development of these network families provides the loop module with the ability to reflect logical facility serving areas and allow for an efficient number of feeder facilities back to the central office. The following provides a detailed description of the methods used by the loop module pre-processing family matching process and the FAMATCH.exe program to create the desired network families and relationships.

Establishing Network Families and Network Sub-Families

Before the FAMATCH.exe program can be run to apply the correct family designations to terminals in the terminalsnap table, a table must be developed that contains a cross-reference of cable ID to family. Before this table is created, however, logic must be applied to all cable counts to determine whether the cable count should be designated as family or subfamily. Upon the completion of the proper family/subfamily designations, the process can proceed with the development of the family table and the FAMATCH.exe program.

VzCost pre-processing defines network families as cable counts originating from all DLCs and crossconnect boxes, and appropriate cable counts originating from the CO MDF. All appropriate cable counts originating from the CO MDF are designated as CO families, while families derived from DLCs and crossconnect box cable counts receive the name of the cable count. For example, suppose a wire center contains C1 and C2 that originate from the CO MDF, Cable AA and Cable C108 that originate from a crossconnect box and a DLC, respectively, and several miscellaneous counts originating from several building terminals scattered throughout the wire center. All serving terminals with a cable count of C1 or C2 would be grouped into the CO family, while terminals with a cable count of AA would form the AA family. Likewise, any terminals with a cable count of C108 would form the C108 family. Any other terminals with a cable count other than C1, C2, AA, or C108 would be classified as subfamilies.

VzCost pre-processing uses the Family table to provide the family and subfamily designation of all cable counts on a wire center basis. The family table uses data from the CPRCOMP table as the source of cable families and subfamilies, and the CIP table as the source of DLC and cross-connect families. The following procedure is used to create the Family table.

Step 1. Create tables Pre-Family and Pre-Subfamily

Structure: Jurisdiction

CLLI

Cable_ID

Family

Subfamily

Category

The Pre-Family and Pre-Subfamily table structure is identical to the Family table structure, with the addition of the category field. The category field is identical to the category field in the CPRCOMP table and is used to limit selections to the pre-family to 'cable' items.

Table contents:

The Pre-Family table will contain:

1. From the CIP table: Records designated with a term_type of 'C' (cross-connect) or 'P' (pair-gain, or DLC).

Table matching: CIP to pre-family and pre-subfamily

Pre-family/Pre-subfam		CIP
Jurisdiction	←	State default
CLLI	←	CLLI
Cable_ID	←	Family
Family	←	Family
Subfamily	←	Null
Category	←	Term_type

2. From the CPRCOMP table: **Distinct** (on CLLI and Cable_id) records where category = 'Cable' and source ipid = '-10' **and cli, cable_id is not already present in Pre-family table** and cable_id <> cable_id on list of ineligible cable counts.

Table matching: CPRCOMP to pre-family and pre-subfamily

Pre-family/Pre-subfam		CPRCOMP
Jurisdiction	←	Jurisdiction or state default
CLLI	←	CLLI
Cable_ID	←	Cable_ID
Family	←	Null
Subfamily	←	Null
Category	←	Category

The Pre_Subfamily table will contain:

1. From the CPRCOMP table – **Distinct** (on CLLI and Cable_id) records where category='Pair gain' and (source IPID = '0' or source ipid='-10')
2. From the CPRCOMP table – **Distinct** (on CLLI and Cable_id) records where category='Cross connect' and source IPID='0'

3. From the CPRCOMP table – **Distinct** (on CLLI and Cable_id) records where category='Cable' and source IPID<>' -10'

Step 2: Pre-family and Pre_subfamily family/subfamily designation

1. In the Pre_family table: Replace family with 'CO' where category = 'Cable'
Note: The family field of the 'P' and 'C' records from the CIP table is already populated with the cable ID of the cross connect or DLC.
2. In the pre_subfamily table: Replace subfamily with Cable_Id.

Step 3: Match pre-family table to pre-subfamily table

Copy records from the pre-sub-family table to the pre-family table where there is not a match on CLLI and Cable ID. This populates the pre-family table with subfamily records, but only if the Cable ID of that subfamily is not already designated as a family in that wire center.

Step 4: Populate Family table

Populate the Family table with fields Jurisdiction, CLLI, Cable ID, family and subfamily from the pre-family table.

At this point the Family table has been completed and contains all Cable Ids and their designation as a family or subfamily. The FAMATCH.exe program is ready to be run.

Description Of Famatch.exe

FAMATCH.exe is a program used to assign appropriate family and subfamily groupings to serving terminals in the terminalsnap table. This allows for logical serving area relationships by providing an accurate reflection of the existing network relationships.

FAMATCH.exe is composed of two processes. The first process, the simpler of the two, assigns family designations to terminals in the terminalsnap table whose counts are designated as families in the Family table. For example, record number 1 in the terminalsnap table might represent a terminal with a CLLI of xxxx and a Cable ID of AA. The FAMATCH program searches the Family table for a CLLI of xxxx and cable ID of AA. When the record is found, FAMATCH.exe examines the family field in the Family table. If the family field is not null, FAMATCH.exe copies the information in the family field of the Family table to the family field in the terminalsnap table. If the family field is null, FAMATCH.exe skips it and goes to the next record in the terminalsnap table. This action is continued until all terminals in the Terminalsnap table with a non-null family field in the Family table have been populated with a Family designation.

The second process examines the remaining terminalsnap records: the ones with no family designation. Suppose record number 2 in the terminalsnap table contains a null family field after the first process is completed. FAMATCH.exe examines the cable ID

of the record and searches for a record in the Family table that matches on CLLI and Cable ID. It copies the subfamily field of that record to the subfamily field in the terminalsnap table, much as the first process copied family designations to other records. At this point, the second process goes a step further. It searches the terminalsnap table for the nearest (geographically speaking) terminal in the appropriate wire center with a family designation and copies its family designation to the terminal record under study. This process produces a logical grouping of terminals in defined areas and paths miscellaneous (subfamily) cable Ids to a logical, efficient point.

5.3.3 Spannet.exe

Spannet is the last of the programs in the preprocess and is used to create the Network table from the pre-network table. It is run after the FAMATCH and after the data from the Terminalsnap table is transferred to the Prenetwork table with the snaptonet.qry.

At this point in the process, the Prenetwork table consists of terminal and wire center data only. Before the Spannet.exe is run, cross connect, DLC, and control point data is uploaded from the CIP table to the Prenetwork table.

Items are distinguished in the term_type field of the Prenetwork file, as follows:

ITEM TERM_TYPE designation

Terminal F
Cross connect C
Pair gain (DLC) P
Control point I
Wire center W

The Spannet.exe uses the location and family data of all items in the Prenetwork table to develop a logical distribution and feeder network. Terminals are connected with a spanning process and grouped according to their family designation. Terminals with a CO family designation are routed to the nearest control point while non-CO family terminals are routed to their appropriate cross connect or DLC location, which in turn is routed to a control point. Once routed to a control point, the control point schematic, spanned from control point to control point, serves as the routing back to the central office.

To provide a forward looking view, Spannet.exe also converts, based on distance, certain terminals and control points to **possible** DLC locations, called T's. Logic in the loop module then evaluates each 'T' and determines whether it should be converted to a DLC.

Items are distinguished in the termtyp field of the Network file, as follows:

ITEM TERMTYPE designation

Feeder terminal	1
Distribution terminal	2
Cross connect	X
Pair gain (DLC)	F
Control point	I
Wire center	W
TPOI	T

6.0 Post Network table activities

Following completion of Spannet, the Network table is available for input into VzLoop. At this point in the process, a final demand reconciliation is performed between the Demand table and the Network table. All demand is stripped from the Network table and repopulated with data from the Demand table. The Demand table includes all demand, including switched, DSL, non-switched, and DS1 demand, and populates the Network table based on matches of CLLI code and terminal ID.