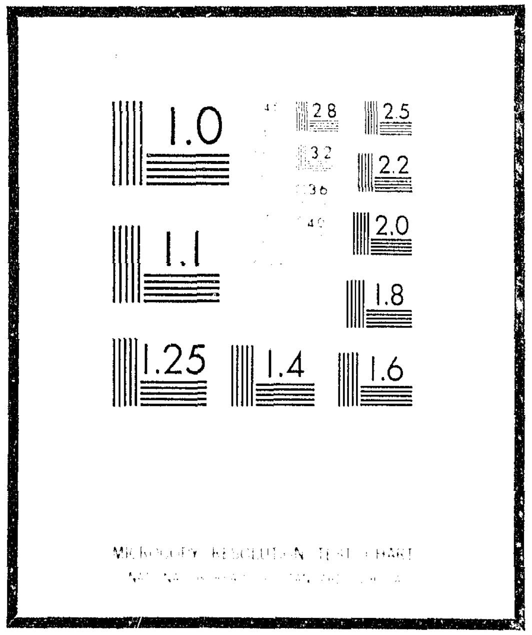


PB-239 532

CHARLOTTE USAC PROJECT
POLICE SYSTEM LEVEL SOFTWARE DESIGN SPECIFICATION

CITY OF CHARLOTTE, NORTH CAROLINA

PREPARED FOR

DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT

OCTOBER 1974

25232

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

BIBLIOGRAPHIC DATA SHEET	1. Report No. USAC CNCO-0168	2.	3. Recipient's Accession No. PB 239 532

4. Title and Subtitle

CHARLOTTE USAC PROJECT
POLICE SYSTEM LEVEL SOFTWARE DESIGN SPECIFICATION

5. Report Date

October, 1974

6.

7. Author(s)

8. Performing Organization Rept. No.

MIS-4311/011/00

9. Performing Organization Name and Address

City of Charlotte
USAC Project
Charlotte, North Carolina 28204

10. Project/Task/Work Unit No.

11. Contract/Grant No.

H-1216

12. Sponsoring Organization Name and Address

U. S. Department of Housing and Urban Development
USAC Project
451 - 7th Street, S. W.
Washington, D. C. 20401

13. Type of Report & Period Covered

Initial Task Completion Report

14.

15. Supplementary Notes

16. Abstracts

The report is from a USAC series produced by the City of Charlotte, covering activities from systems analysis through implementation and evaluation of urban information systems. The nature of the information handled by the Police Department requires special considerations concerning security, interfaces with other systems and response time to transactions entered by Police Department dispatchers without interfering with processing requirements levied against the City's computer system by other IMIS modules. Therefore, front end processing hadrware and software were acquired to provide the special handling requirements. This document presents the rationale for and description of the Front End Processor (FEP) Module, developed under the IMIS project for the Police Department of Charlotte, North Carolina.

17. Key Works and Document Analysis. 17a. Descriptors

- * Information systems,
- * Urban areas,
- * Local government,
- * Data processing,
- * Computer systems software.

17b. Identifiers/Open-Ended Terms

- * USAC (Urban information sys.ems interagency committee),
- * Municipal information systems,
- * Charlotte (North Carolina),
- * Total system,
- * Systems design.

17c. COSATI Field/Group 05 B

18. Availability Statement

Released for distribution by NTIS.

19. Security Class (This Report)

UNCLASSIFIED

20. Security Class (This Page)

UNCLASSIFIED

21. No. of Pages

0 62066

MIS-4311/011/00

USAC CNCO-0168

POLICE SYSTEM LEVEL SOFTWARE

DESIGN SPECIFICATION

October, 1974

Contract No. H-1216

DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT

Office of the Assistant Secretary for Research and Technology

CITY OF CHARLOTTE
600 East Trade Street
Charlotte, North Carolina 28202

This is one report in a series based on efforts pertaining to computerbased municipal information system research and development. The reports will cover the areas of systems analysis, conceptualization, design, development, implementation, evaluation, orientation and training, inter-USAC transfer, and computer systems software.

All reports will be the result of research and development projects sponsored by a consortium of Federal agencies working through the Urban Information Systems Inter-Agency Committee (USAC) under the leadership of the Department of Housing and Urban Development (HUD). Neither the United States Government, nor HUD/USAC, nor any person acting on behalf of HUD/USAC:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method or process disclosed in this report.

As used in the above, "person acting on behalf of HUD/USAC" includes any employee or contractor of HUD/USAC, or employee of such contractor, to the extent that such employee or contractor of HUD/USAC, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with HUD/USAC, or his employment with such contractor.

USER COMMENT FORM

This document is only one of a series describing Integrated Municipal Information Systems. Its title is ______________________________ City

Your comments, pro and con, would be greatly appreciated in assisting in the evaluation and improvement of future documentation.

1. How were you informed as to the document's availability? ____________
2. Did the contents of the document meet your:
   Expectations? -- ____________
   Requirements? -- ____________
3. Was the organization of the document:
   Logical? --- ____________
   Easy to follow? -------------------------------------- ____________
4. Was the document:
   Clear? --- ____________
   Complete? -- ____________
   Consistent? -- ____________
5. Were the non-textural materials complementing the text presented in an understandable manner? ------------------------------------ ____________
   Figures? --- ____________
   Charts? -- ____________
   Others? -- ____________
6. Did you find the quality of reproduction of the document acceptable? ____________
7. Was the timeliness of receipt of the document satisfactory? ____________
8. Please give the name of your department and the type of organization to which you belong. ____

   ____

Please comment further on any items covered or not covered in the list. Use Additional Pages if Necessary.

If you wish to receive a complete list of present and future documentation, please furnish your name and address below:

Name______________________________ Date____________________

Address____

Please return form to:
USAC Documentation Working Group Chairman
A146 Technology Building
National Bureau of Standards
Washington, D.C. 20234

TABLE OF CONTENTS

1. INTRODUCTION

Early in the Integrated Municipal Information System (IMIS) development effort it became apparent that the nature of the information being handled by the Police Department would require special consideration from the point of view of security, interfaces with other systems, and response time to transactions entered by Police Department dispatchers without interfering with the processing requirements levied against the City's computer system by other IMIS modules. Therefore, a decision was made to acquire front end processing hardware and software that would provide the special handling required.

The hardware obtained was a Burroughs B2700 system. The software was developed by the IMIS staff using Network Definition Language (NDL) and FORTE packages supplied by Burroughs.

This document presents the rationale for and description of the Front End Processor (FEP) Module, developed under the IMIS project for the Police Department of Charlotte, North Carolina.

Specifications for modules that interface with the FEP are not included in this document, but may be found in their respective module specification document.

It is not the scope of this document to enter into discussion the concepts of any software supplied by Burroughs Corporation. For more information, refer to the following documents:

Burroughs Medium Systems COBOL Reference Manual (1063708)

Burroughs Medium Systems Network Definition Language (NDL)[1]

Burroughs Medium Systems File Organization Techniques (FORTE)[1]

Burroughs Advanced Assembler and System Software Manual (1055969)

Burroughs Medium Systems Software Operational Guide (MCP) (1054772)

[1] Documentation supplied on magnetic tape with each software release.

2. FEP REQUIREMENTS

The requirements of the FEP design were imposed in part by the local environment in which the Police Department must operate, and in part by other agencies with whom they must interface. The following paragraphs discuss these and other requirements.

2.1 Criteria For Selecting the FEP

2.1.1 Response Time

One of the first IMIS modules considered for development for the Police Department was the Field Assignments Module. It provides an automated dispatching capability and if it is to be effective, must respond rapidly to the operators' inquiries. Response time for this module was set at five seconds. Experience with the City's existing system indicated such a short response would not normally be possible. It was felt that as more and more modules were implemented on the City system, the contention for operating space and core processing time would continue to slow response to Police Department inquiries until an unacceptable condition was reached.

Only a system dedicated to this special task would meet the requirement. The principal use of the B2700 will be the support of the Field Assignments Module with the major portion of the module's programs operating in the B2700. It is anticipated that this design will provide the required response time.

2.1.2 Security

Law Enforcement Persons and Records Index Module (scheduled for implementation at a later date) has a requirement to obtain data from and transmit data to the state Police Information Network (PIN) in Raleigh, North Carolina, and hence to the FBI's National Crime Information Center (NCIC) in Washington, D. C. The agencies controlling these systems allow another system to join the network only if its computer system is under direct control of a criminal justice agency. The lines to these state and federal systems will connect with the FEP and prevent unauthorized communication with PIN/NCIC.

One of the facilities provided by the Field Assignments Module allows dispatchers to obtain an up-to-the-minute status of police field units, including their location. The nature of this sensitive data makes security a design requirement. By providing a dedicated computer system within the Law Enforcement Center Building and thus eliminating outside lines, it was felt that an acceptable measure of securi.y could be provided which would not be possible under other conditions.

The FEP software provides a measure of security by allowing specific transactions from only certain terminals. Software insures that test or training terminals do not update operational data. Lockout capabilities exist within the FEP which prevent multiple programs from simultaneously updating the same data segment.

2.2 Criteria For The FEP Design

2.2.1 IMIS Modules Supported

The FEP must support the following modules to the degree that their requirements are known at the time the FEP was designed:

- Field Assignments
- Law Enforcement Persons and Records Index
- Deployment Planning
- Investigative Operations

2.2.2 External Interfaces

The FEP must interface with the following:

- TD800 Video Terminals
- City Data Processing System
- PIN/NCIC Systems
- County System (This interface is not an immediate requirement, but rather a projected one.)

2.2.3 Core Allocation

The B2700 has 150K bytes (maximum) available. The initial allocation is based upon estimated sizes of the system components as follows:

Executive — Master Control Program (MCP)[1]	45K
Data Communications Handler	26K
Data Base Handler	26K
Field Assignments Module	23K

[1] The MCP is NOT a part of the FEP, but is the B2700 Executive supplied by, and is proprietary to, Burroughs Corporation.

Free Area 30K
(Used for compilation, utilities and support for other modules.)

The B2700 was acquired mainly to support the Field Assignments Module. It is anticipated, however, that other modules may require programs to operate on the B2700. The 30K free area will be adequate to meet these needs, as they are now known, if care is taken in specifying requirements for FEP support for other modules to be implemented at some future date. Inefficient use of this area may result in system degradation to the degree that major redesign becomes necessary.

The B2700 system has a programmable communications controller which currently has 24K bytes available for use. Within this area must reside interface coding and buffers for TD800's, the City data processing system, PIN/NCIC, and the B2700.

3. FEP DESIGN

The FEP is not a general purpose system. The software developed for the FEP was designed to specifically handle the known requirements of the Charlotte Police Department and the previously mentioned IMIS modules that are to be developed.

Figure 3-1 shows the configuration and interfaces that governed the design constraints.

The TD800 terminals are operated by complaint clerks, dispatchers, records bureau personnel, etc., within the Law Enforcement Center of the Police Department. These operators may query the system for information concerning calls for service, stolen property, criminal histories, etc., and likewise may enter data related to these. The queries enter the FEP where their destination has been programatically determined, and the message processed. Messages may be sent to the City system, PIN/NCIC system, other terminals, or they may be processed by programs in the B2700. In cases where messages are sent to another system, that system will usually generate a display for return to the operator's terminal via the FEP.

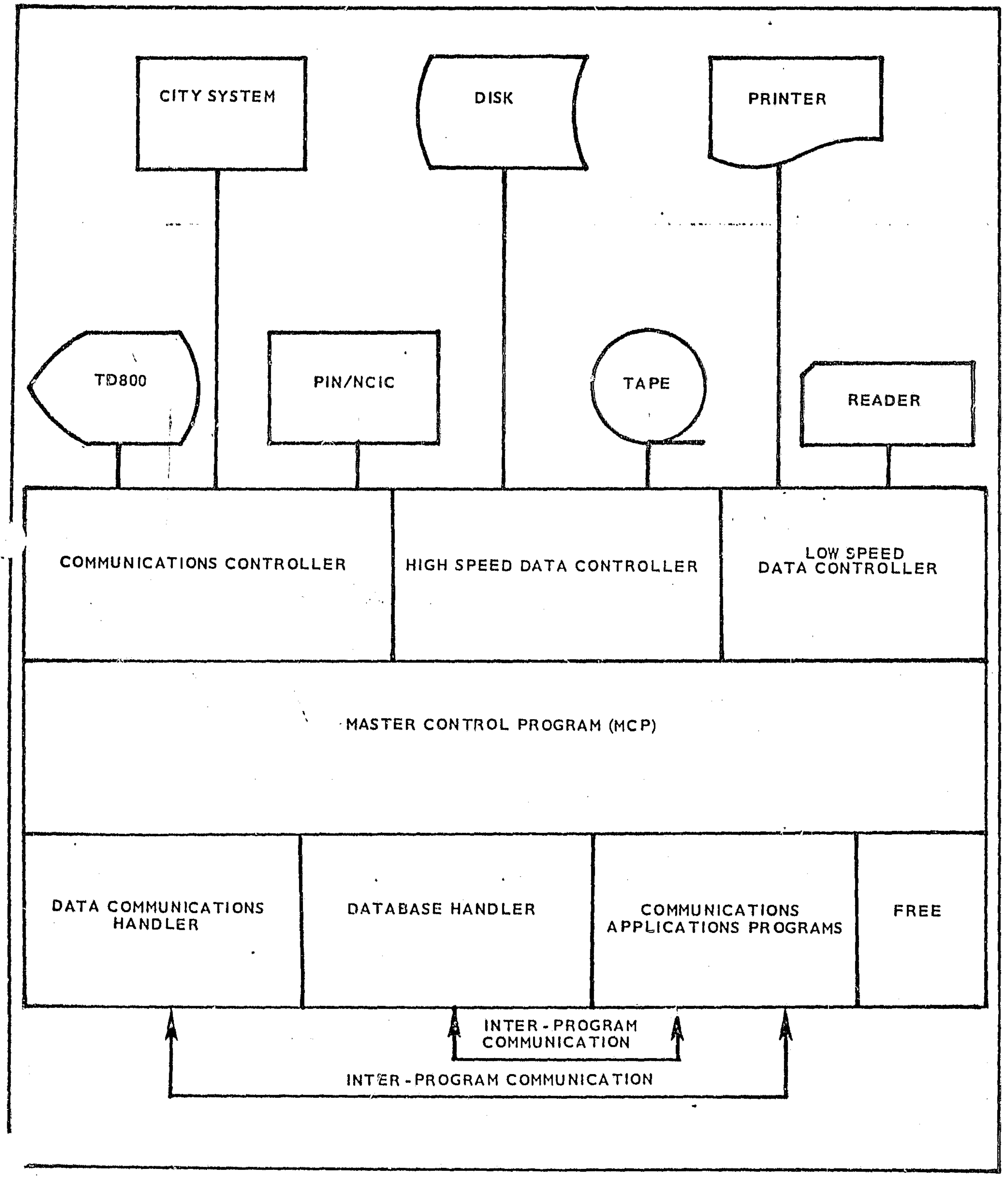
Messages to be processed by the B2700 are passed to the Datacomm Handler for logging, and the message is passed to the application program module (e.g., Field Assignments) responsible for processing it. When the application program needs to access or update the data base, communication is effected with the Data Base Handler. Response messages are passed back through the Datacomm Handler and communications controller to the operator's terminal. In some cases, data may also be sent to another system (i.e., City system) for entry into its files.

3.1 Communications Controller

The software for the programmable communications controller, which handles polling, selecting, message switching, and queueing disciplines, is written using proprietary macros provided by Burroughs Network Definition Language (NDL) and user-coded routines written in Burroughs Assembly Language for transaction code analysis. The first four text characters of all messages received by the communications controller are assumed to be a transaction code by which the destination of the message is ascertained. According to specific transaction code values, the message test is transferred to the B2700, the PIN/NCIC system, or the City data processing system.

3.2 Data Communication Handler

The Data Communications Handler is used to log on tape the messages received from the communications controller and to queue these messages for transfer to the application programs resident within the B2700. In addition, the handler has a major responsibility for system initiation and for recovery following system failure.


CITY SYSTEM
DISK
PRINTER
TD800
PIN/NCIC
TAPE
READER
COMMUNICATIONS CONTROLLER
HIGH SPEED DATA CONTROLLER
LOW SPEED DATA CONTROLLER
MASTER CONTROL PROGRAM (MCP)
DATA COMMUNICATIONS HANDLER
DATABASE HANDLER
COMMUNICATIONS APPLICATIONS PROGRAMS
FREE
INTER-PROGRAM COMMUNICATION
INTER-PROGRAM COMMUNICATION

FEP CONFIGURATION AND INTERFACES
FIGURE 3-1

3.3 Data Base Handler

FORTE, an acronym for File Organization Techniques, is the software used for data base accessing and maintenance for all programs. It will open all files when the system is brought up and close them at shutdown.

When the data base is to be modified, the calling program informs the Data Base Handler to lock out the accessed segments so that no other programs may update those segments until the modification is complete. If the modification is not completed within eight seconds, the segment is unlocked automatically, so that other programs are not locked out indefinitely; the requested modification is ignored and no change is made in the data base.

3.4 Initiatization

The FEP may be initialized in one of three states, each depending upon the conditions under which the system was previously operating. The three states are NORMAL, RESTART, and RECOVERY.

The NORMAL initialization will load the data base from tape and initiate data communications activities. Presuming that the system went through an orderly shutdown the day before, this would be the regular procedure for initialization when the system is brought up again.

The RESTART initialization presumes that the data base is loaded and ready. The FEP immediately initiates data communications activities. This presumes that the FEP was up and running, and then taken down only momentarily without affecting the data base.

The RECOVERY initialization is used after an unrecoverable system error has occurred, and the only recourse is to start from the beginning. The RECOVERY will remove the old data base and reload the data base from tape as it was after the last orderly shutdown. The FEP log tape is also copied to disk, and RECOVERY begins by reprocessing the logged transactions. After this is complete, normal data communications are initiated and incoming messages again logged.

3.5 Shutdown

Each application module may be shut down by its module control terminal (e.g., the watch supervisor for Field Assignments), thereby allowing modules to operate independently of each other. A module will go through an orderly shutdown saving whatever data is pertinent to its next run, then go off-line. After all modules are shutdown, the communications network may be brought down by computer operator intervention. At this point the Data Base Handler will go to end-of-job after all batch programs which interact with it have terminated. The data base will be dumped to tape (as a checkpoint), and the Data Base Handler will terminate.

Should it be necessary to access the data base while the data communications system is not in use, the Data Base Handler may be re-initiated. Following its use, however, the data base should be dumped again. Only the most recent data base dump tape should be reloaded at NORMAL initialization time.

3.6 Recovery

To support recovery in the event of system failure, all transactions entering the B2700 are logged on tape. For recovery processing, it will be necessary to reload the data base from the previous data base dump tape and reprocess the transactions on the log tape. In the recovery process, the log tape will be copied to disk by the Datacomm Handler while the data base is reloaded. Then the log will be processed and a new log tape created containing the old transactions and any new transactions entered following completion of the recovery. No new transactions are accepted until all old transactions are completed.

Individual modules will be responsible for specifying manual procedures to be followed while the system is inoperative or recovering from a failure. *Note:* Updates to the data base made by batch programs are not recorded on the log tape and require being rerun in event a recovery is necessary.

3.7 Training Mode

The Police Department requires that the FEP support a training mode whereby terminal operators may take actions as though they were operating in a live mode. However, their actions must not affect the operational data base.

To effect this requirement, an operator may place his terminal in the training mode using a special transaction code. He may also specify the position his terminal is to simulate. The operator may remove his terminal from the training mode in a similar manner. (A terminal can operate in only one mode at any given time — operational, training, or testing.) In general, a program need not be aware of whether or not the terminal it is servicing is in the training mode. The FEP will access the correct data base and prevent an update of operational files.

Training terminals interact with the data base in the following manner. When the operator queries the data base, a pseudo data base is checked first to see if data exists that meets the requirements of the inquiry. If so, it is returned to the calling program. If not, the

operational data base is checked; and the qualifying data is returned to the calling program. If the operator takes an action which would normally result in an update to the data base, the pseudo data base is updated instead.

4. TRANSFERABILITY

The Front End Processor Module is specifically designed to support Charlotte IMIS Police Department modules on a B2700 system within the unique environment of this municipality. The utility of transferring the FEP to another environment must be carefully judged by the recipient organization.

The communications computer is programmed in Burroughs Advanced Assembly Language which could hinder transferability. The Datacomm and Data Base Handlers are written in COBOL, which should facilitate transfer of these programs.

Although the programs themselves may not be easily transferable to another environment, the design concept should be readily adaptable to most environments requiring the previously described data communications and data base handling capabilities.

5. DETAIL DESIGN CONSIDERATIONS

The following paragraphs provide additional information concerning programming requirements and considerations.

This document is not intended to fully satisfy the detailed technical informational needs of programmers assigned to development/maintenance of either the FEP systems level software itself or any applications level (IMIS modules) software directly interfacing with the FEP.

For such detailed information, applications level software programmers must become familiar with the *Police Field Assignments User Manual, Program Maintenance Section* (MIS-7601/016). FEP systems level software programmers must become familiar with the *Burroughs Medium Systems Network Definition Language Manual* and the *Burroughs Medium Systems Disk Forte Manual.*

5.1 Communications Controller

In order to provide to components of the Police system the address of the terminal to which a message must be sent, all messages to the City system will have BEL/BEL/AD1/AD2/ETX as the last five characters of each message. The AD1 and AD2 indicate two characters which constitute the logical address of the intended receiving terminal. If the message originates in the B2700, the Datacomm Handler will generate these characters; if from a terminal, the Communications Controller will generate them. All City system programs interfacing with the B2700 must provide the same characters for re-routing replies. See Appendix A for specific logical addresses.

As new modules that must interact with the Communications Controller are developed, it will be necessary to provide the controller with the respective module transaction codes and the destination of these codes (i.e., City system, PIN/NCIC, B2700, etc.). This will require modification to tables internal to the controller. (See Appendix B for a list of currently supported transaction codes.)

Each line has a buffer for its messages in the Communications Controller. One additional buffer is allocated for all lines to share to improve throughput. The buffers for the TD800 are large – 1920 bytes. This, and the limitation of a maximum core size of 32K, requires careful attention to communication requirements that affect program and terminal buffer size.

The NDL for the Communications Controller permits only the B2700 to act as a master system. All other terminals and/or computers communicating with the B2700 must do so on a slave basis.

5.2 Data Communication Handler

The Datacomm Handler provides the means by which messages are logged and queued for transfer to an application program. The method of interface between the program and the handler will utilize the COBOL FILL verb, a core-to-core transfer of data from one program to another. The executive software, the MCP, will effectively MOVE PROG-1-DATA TO PROG-2-DATA. Using this method, each program can operate independently of the other until one needs the services of the other – the applications program operates until it needs the service of the handler, the handler operates until it needs to service the applications program. The following is an example of the interface between the application program and the Datacomm Handler.

Definition of Datacomm-Handler-to-Application-Program Transfer Area

```
01    NDL-INTERFACE

      03    PROGRAM-ID                  X(6).
      03    SYSTEM-STATUS               9         COMP.
      03    TERMINAL-NUMBER             99        COMP.
      03    TERMINAL-STATUS             9         COMP.
      03    TERMINAL-TYPE               99        COMP.
      03    MESSAGE-LENGTH              9(4)      COMP.
      03    MESSAGE-TIME                9(8)      COMP.
      03    PROGRAM-SWITCH              X(6).
      03    TERMINAL-SWITCH             99        COMP.
      03    TERMINAL-REPLY-SWITCH       99        COMP.
      03    TERMINAL-CONSTANT           X(8).
```

```
      03    MESSAGE-TEXT                X(n) where 1 < n < 1921.
```

The following is a description of the interface area.

- *NDL-INTERFACE* is the record name of the data area to be passed to or from the Datacomm Handler by the COBOL FILL verb.

- *PROGRAM-ID* contains the six-character name of the Datacomm Handler when the message is passed to the applications program. The applications programmer must place the name of his program in this area *before* passing a message to the Datacomm Handler. If this item equals GO-OFF, then the applications program should go through an orderly shutdown, and not communicate with the Datacomm Handler again.

- *SYSTEM-STATUS* indicates whether the system is in an operational or recovery mode of operation.

- *TERMINAL-NUMBER* contains the number of the terminal from which the message originated. This field must not be modified since it is used by the system for linkages.

- *TERMINAL-STATUS* indicates whether the terminal is operational, in a training mode, or in a test mode.

- *TERMINAL-TYPE* defines the type (code) of terminal which sent the message.

- *MESSAGE-LENGTH* contains the count of characters following the STX, including the ETX. It need not be sent by the programmer for outbound messages to terminals.

- *MESSAGE-TIME* contains the time the message was received, in milliseconds.

- *PROGRAM-SWITCH* must contain the program name of the intended receiving program with which Datacomm must communicate when it is desired to send a message to another program.

- *TERMINAL-SWITCH* contains the logical address of the intended receiving terminal when it is desired to send a message to a terminal other than the originating terminal.

- *TERMINAL-REPLY-SWITCH* must contain the number of the terminal to which the City computer will reply, after this message has been processed by the City system. If not communicating with the City system, this area has a value of zero.

- *TERMINAL-CONSTANT* is an eight-byte field used to hold any constant which may be desired for programmatic use. This constant is unique for each terminal.

- *MESSAGE-TEXT* contains up to 1920 bytes (inclusive of the ETX) used as a message area for transmission to terminal.

The following example program will accept a message from a terminal, display it on the system console, and echo the message back to the terminal.

```
IDENTIFICATION DIVISION

PROGRAM-ID. EKKKKO.

ENVIRONMENT DIVISION

DATA DIVISION

WORKING-STORAGE SECTION.

01    NDL-INTERFACE.
      03    PROG-ID             PIC     X(6).
      03    SYS-STAT            PIC     9          COMP.
      03    TERM-NO             PIC     99         COMP.
      03    TERM-STAT           PIC     9          COMP.
      03    TERM-TYPE           PIC     99         COMP.
      03    MSG-LGTH            PIC     9(4)       COMP.
      03    MSG-TIME            PIC     9(8)       COMP.
      03    PROG-SW             PIC     X(6).
      03    TERM-SW             PIC     99         COMP.
      03    TERM-REPLY          PIC     99         COMP.
```

```
      03    TERM-CONST          PIC     XXXX.

      03    MESSAGE             PIC     X(50).

PROCEDURE DIVISION.

START.

      DISPLAY "-- I'LL SHOW YOU THE NEXT MESSAGE".

      FILL NDL-INTERFACE FROM "POSNDL".

      IF PROG-ID = "GO-OFF" THEN STOP RUN.

      DISPLAY "--" TERM-NO "SAID" MESSAGE.

      DISPLAY "--" NOW I'LL SEND IT BACK".

      MOVE "EKKKKO" TO PROG-ID.

      FILL NDL-INTERFACE INTO "POSNDL"

      GO TO START.
```

In the above example, the Datacomm Handler would execute the program; and when it was ready to receive the message (the first FILL), the data would be placed in NDL-INTERFACE. The program would process the message and send it back to the terminal (second FILL). At Datacomm generation time, the user could specify that EKKKKO could be rolled out to disk after processing the message, or it may have been made core-resident, waiting for the next message.

5.3 Data Base Handler

The Data Base Handler is driven by function requests issued by application programs. These requests may include GET, GETH (get and hold for update), REPL (replace after update), ISRT (insert a new segment), DLET (delete an existing segment), and FREE (release a held segment). Each time it requests an action, the program must specify which data base segment it will be accessing and its key. The following is an example of the FORTE interface between the application program and the Data Base Handler.

```
01   FORTE-INTERFACE

     03   PROG-ID                PIC     X(6).

     03   OVLY-ID                PIC     X(6)

     03   FUNCTION               PIC     XXXX.

     03   SEGMENT-ID             PIC     XXXX.

     03   SEG-KEY                PIC     X(16).

     03   FORTE-REPLY.

          05                     STATUS PIC     XX.

          05                     FILLER-RESERVED

          05                     SEGMENT-DATA  PIC
```

The following is a description of each of the function calls accepted by the handler:

- *GET* is used to access a segment without updating it. All updates must use the GETH function.

- *GETH* is used when updating is required. The GETH will lockout the segment type specified by the SEGMENT-ID, so that no other program may interfere while the updating is taking place.

  If the segment is not unlocked within eight seconds by a FREE, DELT, REPL, or another GETH from the same program, the handler will unlock the segment automatically. While a segment type is locked, it may still be accessed by other programs using the GET function. An ISRT may also be used under lock conditions.

- *ISRT* is used to insert a new segment into the data base. The Data Rase Handler does check for a duplicate (based on the key) before inserting the new segment.

- *REPL* replaces the segment retrieved by the last GETH for that segment type. The REPL will unlock the segment, so that a FREE will not be required. If a GETH was not performed before the REPL, the handler will notify the program of the error.

- *DLET* allows a segment to be deleted from the data base, provided the segment was not locked by another program.

- *FREE* frees a segment locked by a GETH request.

Note: To facilitate transferability all data base functions performed by the Data Base Handler are coded in COBOL.

The following is an example of a program which would access the data base to delete a specific segment based upon parameters from the console.

```
IDENTIFICATION DIVISION

PROGRAM-ID. DELETE

ENVIRONMENT DIVISION

DATA DIVISION

WORKING-STORAGE SECTION.

01   FORTE-INTERFACE

     03   PROG-ID                    PIC     X(6)     VA "DELETE".

     03   FILLER                     PIC     X(6)     VA SPACES

     03   FUNCTION                   PIC     XXXX.

     03   SEG-ID                     PIC     XXXX.

     03   SEG-KEY                    PIC     X(16).

     03   FORTE-REPLY

          05   STATUS                PIC     XX.

          05   FILLER                PIC     XXXX.

          05   SEG-DATA              PIC     X(150).

PROCEDURE DIVISION.
```

```
START.
ACCEPT SEG-ID.
IF SEG-ID = "STOP" THEN STOP RUN.
ACCEPT SEG-KEY.
MOVE "GETH" TO FUNCTION.
FILL FORTE-INTERFACE INTO "POS4TE".
FILL FORTE-REPLY FROM "POS4TE".
IF STATUS NOT = "OK"
        DISPLAY "-- SORRY, NOT THERE"
        GO TO START.
MOVE "DLET" TO FUNCTION.
FILL FORTE-INTERFACE INTO "POS4TE".
FILL FORTE-REPLY FROM "POS4TE".
IF STATUS NOT = "OK"
        DISPLAY "--WON'T GO AWAY".
GO TO START.
```

PARAMETER REQUIREMENTS

Function	Segment Name	Segment Key	Record Area
GET	P	P	--
GETH	P	P	--
ISRT	P	P	P
DLET	P	P	–
REPL	P	–*	P
FREE	P	–	--

P = Programmer must supply this data.

– = Not used.

* = Always replaces the segment retrieved by the last GETH for that segment type.

STATUS CONDITIONS RETURNED

GET = OK, NO-RECORD, SYSTEM-ERROR

GETH = OK, NO-RECORD, SYSTEM-ERROR

ISRT = OK, DUPLICATE, FILE-FULL, ** SYSTEM-ERROR

DLET = OK, NO-RECORD, SYSTEM-ERROR

REPL = OK, SYSTEM-ERROR

FREE = OK, SYSTEM-ERROR

**On a FILE-FULL condition, the record has been added to the file; and everything is normal. However, any attempt to add more segments to the file will result in a SYSTEM-ERROR. *NOTE:* Some files contain more than one segment type. FF is related to the file, not the segment type. (Appendix C lists currently available data base segments.)

APPENDICES

APPENDIX A

LOGICAL AND PHYSICAL RELATIONSHIPS OF TERMINAL IDENTIFIERS

The following table shows the logical and physical relationships of terminal identifiers. Field Assignment Position Identifiers are the codes used by applications programs to determine the origin of a message. Programs operating in the B2700 which communicate with an external device will supply Datacomm with the logical identification of the device to which a message is to be sent. Physical addresses are placed in messages sent to devices by the Communications Controller.

Logical Terminal Number	Physical Address	Position	Comments
01	DC	DC	Data Communication Controller
02	OA	PINC	Police Information Network Control
03	1A	MECK	Mecklenburg County Computer
04	2A	CHAR	City of Charlotte Computer
05	3A	SCAR	South Carolina Computer (as yet undefined)
06	4A	C1	Field Assignments Clerk Number 1
07	4B	C2	Field Assignments Clerk Number 2
08	4C	C3	Field Assignments Clerk Number 3
09	5A	D1	Field Assignments Dispatcher Number 1
10	5B	D2	Field Assignments Dispatcher Number 2
11	5C	D3	Field Assignments Dispatcher Number 3
12	5D	R1	Field Assignments Records Clerk Number 1 (as yet undefined)
13	5E	CR	Code Room (as yet undefined)
14	5F	HD	Hot Desk (as yet undefined)
15	4D	T1	Training Terminal Number 1
16	4E	T2	Training Terminal Number 2
17	SP	SP	Data Communications SPO

Preceding page blank

Messages sent to the B2700 from the City computer must use the address to identify the destination, if different from the originating terminal.

APPENDIX B

TRANSACTION CODES (Currently Serviced)

This appendix should be updated whenever new transaction codes are added to FEP.

The following transaction codes are currently recognized by the Message Switcher.

Transaction Number	Transaction Name	Destination
SC01	Display/Establish Unit Status Data	B2700
SC02	Activiate Unit Status	B2700
SC03	Generate Call/Assignment Record	B2700
SC04	Assign Unit(s) to Call	B2700
SC05	Update Unit Assignment Status	B2700
SC06	Update Call/Assignment Active Data	B2700
SC07	Assign Additional Units	B2700
SC08	Generate Assignment Summary	B2700
SC09	Generate Organization/Dispatcher Assignment Summary	B2700
SC10	Display/Update Dispatcher Position Status	B2700
SC11	Review/Update Tactical Response Plan	B2700
SC12	Update Front End Processor	B2700
SC13	Request List of Delinquent Field Reports	City System
SC14	Update Call/Assignment History Data	B2700

Preceding page blank

SC15	Update Call/Assignment Counter	B2700
SC20	DSC02 Returned	B2700
SC21	DSC05 Returned	B2700
SC22	Units Assigned by Dispatcher	B2700
SC23	Display Alternate Units	B2700
SC24	Get TRP From POC505	B2700
SC25	Error in Assignment Summary Input	B2700
SC26	Purge C/A History Records From Police System	B2700
SC27	Update DSC16	B2700
SC30	Generate Assignment Summary	City System
SC31	Add/Modify C/A History Record	City System
SC91	Assign Field Assignments Training Terminal	B2700
SC98	Initiate Shutdown	B2700
SC99	Complete Shutdown Field Assignments	B2700

APPENDIX C

SEGMENTS CURRENTLY ACCESSABLE BY THE DATA BASE HANDLER

This appendix should be updated whenever new segments are made available to the Data Base Handler.

The following segments are currently accessable by the Data Base Handler.

Segment	File	Description
S188	PFSF1	Phonetic Street Name
S189	PFSF1	Phonetic Place Name
S190	PFSF1	Police Street Number
S191	PFSF2	TRP Description
S192	PFSF2	Remarks Data
S193	PFSF2	TRP Time Difference
S194	PFSF3	Alert Messages
S195	PFSF3	Previous Calls
S196	PFSF4	Intersections
S197	PFSF7	Call Assignments—Active
S198	PFSF6	Dispatcher Data
S199	PFSF6	Dispatcher Open Calls—Overflow
S200	PFSF5	Field Units
S201*	PFSF7	Field Assignments Systems Data
S202	PFSF8	Response Area
S203	PFSF5	Field Units By Team
S204	PFSF7	Call Assignments—Units

Preceding page blank

*Access is restricted to POCFOO/System.

END