REPORT NO. DOT-TSC-OST-73-35

A CARGO DATA MANAGEMENT DEMONSTRATION SYSTEM

Juris G. Raudseps
Robert S. Tinkham

FEBRUARY 1974

FINAL REPORT

Prepared for
DEPARTMENT OF TRANSPORTATION
OFFICE OF THE SECRETARY
OFFICE OF THE ASSISTANT SECRETARY
FOR
SYSTEMS DEVELOPMENT AND TECHNOLOGY
Washington DC 20590

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT-TSC-OST-73-35		

4. Title and Subtitle	5. Report Date
A CARGO DATA MANAGEMENT DEMONSTRATION SYSTEM	February 1974
	6. Performing Organization Code
	8. Performing Organization Report No.
7. Author(s)	
Juris G. Raudseps, Robert S. Tinkham	DOT-TSC-OST-73-35
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Department of Transportation Transportation Systems Center Kendall Square Cambridge MA 02142	OS330/R4603
	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Final Report 7/1/72-6/30/73
Department of Transportation/Office of the Secretary; Office of the Assistant Secretary for Systems Development and Technology* Washington DC 20590	14. Sponsoring Agency Code

15. Supplementary Notes

*Also, Office of the Assistant Secretary for Environment, Safety and Consumer Affairs

16. Abstract

Delays in receipt and creation of cargo documents are a problem in international trade. The work described in this report was performed to demonstrate to interested parties some of the advantages and capabilities of a computer-based cargo data management system. A demonstration system for data management and transmission was assembled at the Transportation Systems Center in Cambridge, MA, tested, and demonstrated. Terminals were installed at a site (Washington, D.C.) remote from the central processor at TSC, with which they communicated by direct distance dialing over telephone lines. The processor prepared tapes for transmission of data to Heathrow Airport, London, via teletype circuits. The tests demonstrated remote data entry, validation, editing, updating, retrieval, privacy protection, and teleprinting of multiple documents from a common data base. This report first broadly describes the technical approach taken and the principal lessons learned. Succeeding chapters describe the capabilities of the demonstration system, specify operating procedures, and document the data structure, the hardware, and the software in detail.

17. Key Words	18. Distribution Statement
Cargo Data, Computers, Data Management	DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22151

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	92	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

PREFACE

This work described in this report was performed in support of demonstrations to industry and government agencies of the capabilities of a cargo data management system. The work was sponsored by the Department of Transportation through the Office of Facilitation, Documentation and Procedures Division and through the Systems Division, Office of Telecommunications. The Documentation and Procedures Division conducted the demonstrations, and the Office of Telecommunication furnished technical direction.

A demonstration system for cargo data management/transmission was assembled, tested and demonstrated. Terminals were installed at a site (Washington, D.C.) remote from the central processor/data at TSC, with which they communicated by direct distance dialing over telephone lines. The processor prepared tapes for transmission of data to Heathrow Airport, London via teletype circuits. The tests demonstrated remote data entry, validation, editing, updating, retrieval, privacy protection, and teleprinting of multiple documents from a common data base.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

This report summarizes work performed at TSC in implementing a simple cargo data management system for the purpose of performing certain tests and demonstrations. The overall project deals with the international transportation of cargo via air. Its purpose is to demonstrate that an automated central data bank tied to a telecommunication network can reduce paper documentation, speed movement of cargo, and serve the data needs of industry and government.

An information management network was assembled with terminals at remote sites and a central processor at TSC. This phase of the DOT test program illustrated data entry and retrieval, generation of a variety of documents, editing for missing data and invalid codes, code transcription, transmission of data for air cargo shipments from the United States to the United Kingdom.

The remainder of this chapter reviews the background and the scope of our technical approach and broadly summarizes the principal lessons learned from this work. Section 2 is a description of the capabilities of the demonstration system with some discussion of future extensions of system capability. Section 3 specifies the procedures for operating the system for the use of an operator having no prior familiarity with the system. Sections 4, 5, and 6 document the data structure, the hardware structure, and the software structure in complete detail.

1.1 PROBLEM AND OBJECTIVE

Delays in receipt and creation of cargo documents are a problem in international trade. The problem is particularly obvious at ports and airports where shipments often pile up awaiting necessary documentation. Present procedures for preparing and processing cargo data are usually manual, slow, repetitious, conducive to error, and expensive. It is hoped that wider use of automatic data processing and rapid data transmission techniques can eliminate these difficulties. The objective of this effort was to prove through cargo data management/transmission (CDMT) tests between the

United States and the United Kingdom that an automated central data bank tied into an electronic transmission system can (a) reduce paper documentation, (b) speed the movement of cargo, and (c) serve the needs of industry and government.

1.2 BACKGROUND

DOT is undertaking the standardization and simplification of the paper work connected with cargo shipments so as to permit computer-controlled information collection and retrieval, validation, and correction, transoceanic data communication, generation of all types of documents, and processing for various summary purposes such as customs control, census reporting, etc. Some members of the cargo transportation community (shippers, carriers, forwarders, brokers, insurers, and government agencies) are already using computer-based data processing systems. A smaller number are applying both computer and data transmission systems to prepare, process and transmit cargo data from one office to another. Some countries have automated or are in the process of automating customs cargo entry procedures. The U.S. Bureau of Customs is developing plans for automation of its cargo entry system throughout the United States. Automated techniques are thus beginning to be employed. However, the systems are mainly "in-house" systems, each having its own classification and coding scheme, and not interfacing with others to permit the automated exchange of data and reduction in cost and number of exchange documents.

1.3 SCOPE OF WORK AND TECHNICAL APPROACH

A simple, automated CDMT system was assembled, tested, and demonstrated. This was the first phase of an evolving series contemplated by DOT, leading to an eventual on-line CDMT utility that will expedite cargo and greatly reduce the paper work of diverse members of the transportation community. To implement Phase I (dealing with air cargo from New York to London) terminals were installed at a remote site communicating by direct distance dialing over the Bell System to a central processor/data base at TSC. The processor prepared tapes for teletype transmission of data to

Heathrow Airport, London. Tests were designed to illustrate:

a. On-line data entry, including editing and user data validation,

b. Updating of data, such as correction of errors and replacement of estimates by actual values,

c. Production of current documents through retrieval of data and printing of bills of lading and commercial invoices in standard form at locations remote from the data entry point,

d. Production of air manifests,

e. Transmission of data to a foreign destination, to print the bill of lading and commercial invoice at the foreign location.

The eventual CDMT utility will be used by competing shippers, forwarders, and carriers, all very conscious of data privacy, who may attempt unauthorized access to each others' data to gain competitive advantage. To serve its purpose, even the initial test bed systems had to be designed with privacy considerations in mind. The integrity of the data base and the ability to detect and apprehend invalid request were demonstrated.

The effort can be divided into four subtasks:

1. Assembly of the CDMT system,

2. Designing and programming of software,

3. Performing the tests and demonstrations,

4. Interpretation of results and planning for future work.

The first subtask consisted of selecting a central processor (an in-house DDP-516), renting data phones for the CPU, and leasing the remote terminals. Minor interface work was necessary to fit the central processor to the data phones.

The second subtask required a precise definition of the scope and content of the data management and transmission tests. Cooperation of DOT Headquarters, TSC, and industry was required and

obtained. Once the task was defined, appropriate software was designed and produced. The computer programs performed the functions of data transmission control, data base management, data entry and editing, document formatting and printing, commodity code look-up, and data security assurance.

Tests and demonstrations comprised the third subtask. The system is available for demonstrations to industry and to government agencies with a minimum of notice and preparation. At least ten to fifteen shipments are in the data base at all times for demonstration, and members of the audience can enter shipments, with the system simulating what would happen if it were following the actual movement of cargo.

The fourth ongoing subtask consists of performing an evaluation of the system. This report documents the system developed and demonstrated at DOT Headquarters on March 21, 1973, with simultaneous transmission of data to the United Kingdom. The scenario of the demonstration is included as an appendix.

DOT contemplates a series of tests, of which this effort is the first. TSC is expected to take an active part in all phases of the tests and in the design, specification, and possibly the procurement of the eventual operating CDMT utility.

1.4 LESSONS LEARNED FROM THE DEMONSTRATION SYSTEM

The development and exercising of the demonstration system brought to light several important matters.

a) Basic feasibility. It is feasible to program a computer to perform the various cargo data management functions: data entry, editing, selective retrieval, generation of multiple documents from a common data base, etc. Privacy of information, with selective access limited to a specific list of participants for each data item, has been demonstrated. The work of the operator can be greatly simplified and made more reliable by types of computer processing and checking which were demonstrated.

b) Communications. Using checks and safeguards which we

built into the demonstration system, it is feasible to perform all of these functions from remote locations, using standard teletype terminals and telephone lines. Transatlantic transmission of the formatted documents was also demonstrated. However, one of the lessons learned is that transmission of documents between computer centers should be in a condensed coded format. This will not only improve efficiency of data transfer but will also circumvent certain technical problems which arise in transmission of precisely formatted documents over standard transoceanic teletype circuits which are "text oriented" rather than "graphics oriented."

c) System flexibility requirement. Due to the number of participants involved in typical transactions and to the range of trade practices, the generation of the data input procedures turned out to be a surprisingly sophisticated and time-consuming process. Although TSC was not directly involved in that effort, it was reflected in an extremely detailed computer program with flexibility for handling a wide variety of special practices and situations. It is reasonable to expect this trend to continue.

2. SUMMARY OF SYSTEM CAPABILITIES

The set of programs to be described were designed and written to provide a working computer cargo data management system for demonstration purposes. The programs produce selected documents used in international trade on teletypes connected through the telephone network to the DDP-516 computer at the Transportation Systems center. The system provides for data entry and editing and permits the reading of individual items of data. Data security is provided by a system of passwords. Provision has been made for generating 5-level paper tapes compatible with the commercial TELEX network for off-line transmission of shipping documents to foreign ports of entry.

The system will print on command either air waybills or commercial invoices for those shipments for which data have been entered. These documents are printed in formats aligned with the US Standard Master Format (Figures 1, 2a and 2b). In addition, air cargo manifests can be prepared in a format consistent with Customs Form 7509 (Figure 3) by specifying a flight number and date and listing the shipments to be included by air waybill number.

Procedures for data input to the system are based on a worksheet form designed especially for this purpose (Figures 4a to 4c). The basic data entry program accepts data items sequentially in the order they are given on the worksheet. The computer types each data field number and accepts the entry into that field. The program accepts input to a number of fields either as full text or as numeric codes (e.g. Dun's codes for commercial firms), which are converted to company names and addresses, product descriptions, etc. for printing on documents. Provisions are made for rejecting certain invalid formats and codes and for immediate error correction by the operator. The program automatically computes various totals, subtotals, and discount amounts that have to be printed on the documents. Once data for a shipment with a given transaction number have been entered, a disk file containing these data is established. This file is interrogated whenever documents pertaining

BILL OF LADING

SHIPPER/EXPORTER GENERAL ELECTRIC CO. INTL SALES DIVISION 159 MADISON AVE. NEW YORK, N.Y.	DOCUMENT NO. BA--76743450 EXPORT REFERENCES SHP REF 247 E61 E74088 353894 AC 18658
CONSIGNEE INTL GEN. ELEC. OF NY LTD. LINCOLN HOUSE 296-302 HIGH HOLBORN LONDON WC1 ENGLAND ATTN B R PICKERING	FORWARDING AGENT - REFERENCES J. D. SMITH INTER OCEAN, INC. 175-41 148 RD. JAMAICA, NEW YORK POINT AND COUNTRY OF ORIGIN USA
NOTIFY PARTY INTL GEN. ELEC. OF NY LTD. LINCOLN HOUSE 296-302 HIGH HOLBORN LONDON WC1 ENGLAND ATTN B R PICKERING	DOMESTIC ROUTING/EXPORT INSTRUCTIONS INTERMEDIATE CONSIGNEE INDUSTRIAL FRT. LONG LANE LONDON AIRPORT LONDON, ENGLAND
PIER OR AIRPORT JF KENNEDY	
EXPORTING CARRIER (Vessel/Airline) BOAC 123 2/26/73	PORT OF LOADING ONWARD INLAND ROUTING
AIR/SEA PORT OF DISCHARGE HEATHROW	FOR TRANSSHIPMENT TO

PARTICULARS FURNISHED BY SHIPPER

MARKS AND NUMBERS	NO. OF PKGS.	DESCRIPTION OF PACKAGES AND GOODS	GROSS WEIGHT	MEASUREMENT
AS ADDRESSED LR 8034 LONDON MADE IN USA	1 CTN	ELECTRICAL EQUIPMENT BTN 85.01 SITC 722.10 GDEST PARTIAL SHIPMENT	POUNDS 1	

These commodities licensed by U.S. for ultimate destination GBR Diversion contrary to U.S. law prohibited.

CHARGEABLE WEIGHT	RATE CLASS	RATE CHARGE	CURRENCY	DECLARED VALUE FOR CARRIAGE	AMOUNT OF INSURANCE
		MIN	US	NVD	

CHARGES	PREPAID	COLLECT
WEIGHT CHARGE		13.75
VALUATION CHARGE		
OTHER CHARGES		
DUE CARRIER		
INSURANCE		
DUE AGENT		HDLG 8.00 GE-SRV 5.00 POST 1.50 INS 2.25
TOTAL		30.50

Figure 1. Air Bill of Lading Form

COMMERCIAL INVOICE

COMMERCIAL INVOICE

EXPORTER
GENERAL ELECTRIC CO. INTL
SALES DIVISION
159 MADISON AVE.
NEW YORK, N.Y.

DOCUMENT NO
INV. NO. 353932

DATE
02/27/73

EXPORT REFERENCES
TRANS NO. TW--76743203
BD 43293 235

CONSIGNEE
INDUSTRIAL FRT.
LONG LANE LONDON AIRPORT
LONDON, ENGLAND

CUST ORD NO. LR84551

FORWARDING AGENT REFERENCES

POINT AND COUNTRY OF ORIGIN
USA

NOTIFY PARTY
INTL GEN. ELEC. OF NY LTD.
LINCOLN HOUSE 296-302 HIGH
HOLBORN LONDON WC1 ENGLAND

DOMESTIC ROUTING/EXPORT INSTRUCTIONS
SEND INVOICE TO.
INTL GEN. ELEC. OF NY LTD.
LINCOLN HOUSE 296-302 HIGH
HOLBORN LONDON WC1 ENGLAND
ATTN B R PICKERING

PIER OR AIRPORT
JF KENNEDY

EXPORTING CARRIER (Vessel/Airline)
TWA 789 3/06/73

PORT OF LOADING

ONWARD INLAND ROUTING

AIR/SEA PORT OF DISCHARGE
HEATHROW

FOR TRANSSHIPMENT TO

ITEM NO.	GROSS WT.	NET WT.	UNITS	DESCRIPTION OF PACKAGES AND GOODS	UNIT PRICE	TOTAL
	POUNDS					
1	14	12	2	45F611FA CAPACITORS PKG 1 11X7X13 IN .58 CU FT	6.75	13.50
2	29	26	8	45F611FB CAPACITORS PKG2 12X18X18 IN 2.25 CU FT	6.48	51.84
				GDEST PARTIAL SHIPMENT		

These commodities licensed by U.S. for ultimate destination GBR Diversion contrary to U.S. law prohibited

PAYMENT TERMS

DELIVERY TERMS
FOB FAC FRT PP TO FIRST DOMES DEST

MARKS & NUMBERS	AMOUNT INSURED	INSURED RATE	CURRENCY
LR 84551 MADE IN USA 2 CTNS 2 PKGS	450		US

FREIGHT AND OTHER CHARGES

INSURANCE CHARGE

TOTAL 65.34

Figure 2. Commercial Invoice Form

ADDENDUM TO COMMERCIAL INVOICE

DOCUMENT NO.	DATE

ITEM NO.	GROSS WT.	NET WT.	UNITS	DESCRIPTION OF PACKAGES AND GOODS	UNIT PRICE	TOTAL

Figure 2. Commercial Invoice Form (Cont'd)

Customs Form 7509
THE DEPARTMENT OF THE TREASURY
6.7, 6.8, 6.9, C. R.
Mar. 1969

AIR CARGO MANIFEST

BUREAU OF CUSTOMS

Form approved
Budget Bureau No. 48 R0152

Owner or operator TWA

Marks of Nationality and Registration* USA Flight No. 789 Date 3/06/73

Point of lading JF KENNEDY INTL. AIRPORT (Place) Point of unlading LONDON HEATHROW AIRPORT (Place)

AIR WAYBILL NO	NUMBER OF PACKAGES	NATURE OF GOODS*	FOR USE BY OWNER OR OPERATOR ONLY	FOR OFFICIAL USE ONLY
TW--1[illegible]721813	1 CASE	MEDICAL X-RAY EQUIPMENT		
TW--64244832	1 CTN	MEDICAL ELECTRICAL EQUIPMENT		
TW--74559273	29 DRUMS	SYNTHETIC RESIN		
TW--85820324	2 CTNS	DIAGNOSTIC PRODUCTS		
TW--85820980	1 CTN	DIAGNOSTIC EQUIPMENT		

Figure 3. Air Manifest Form

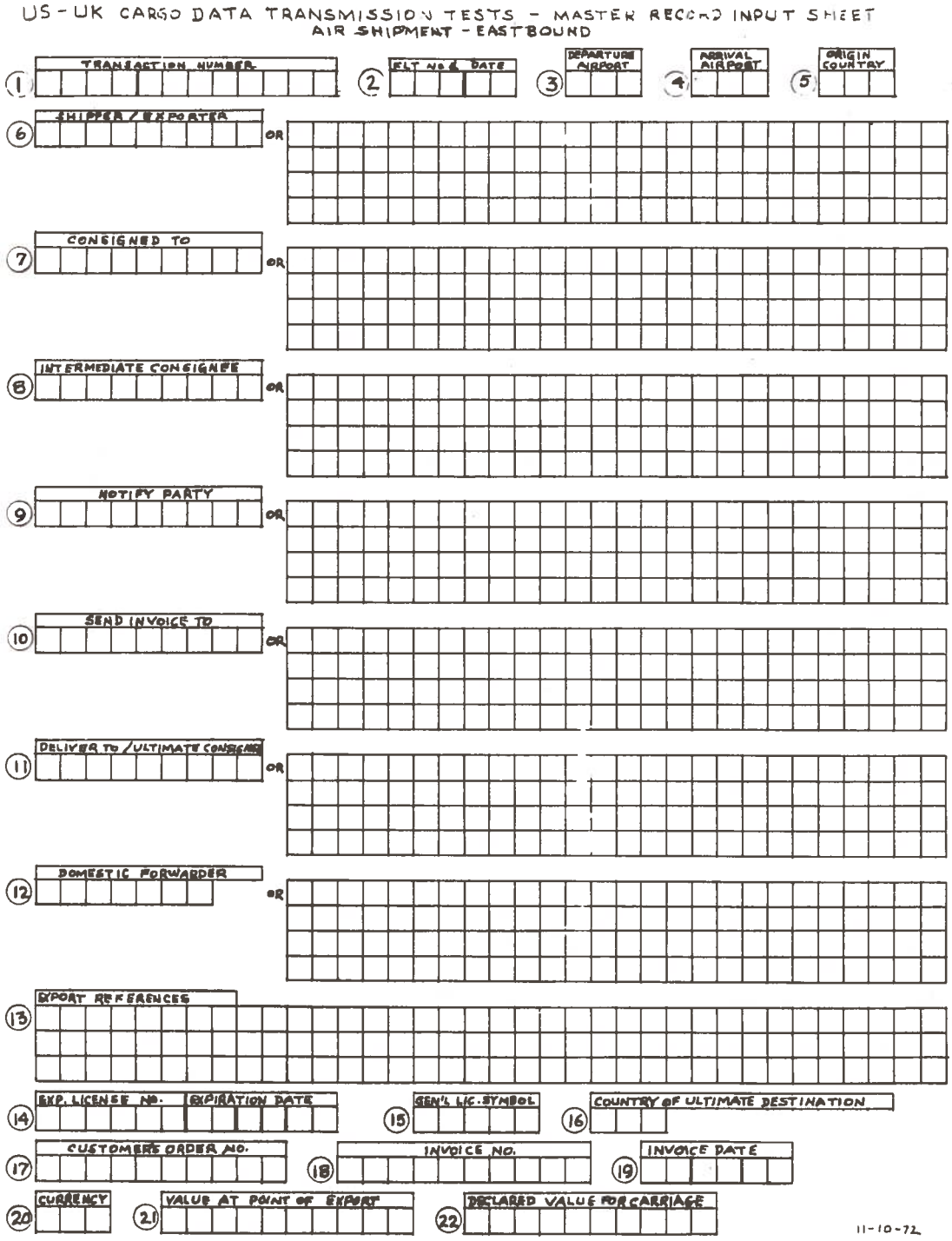

Figure 4. Data Input Form

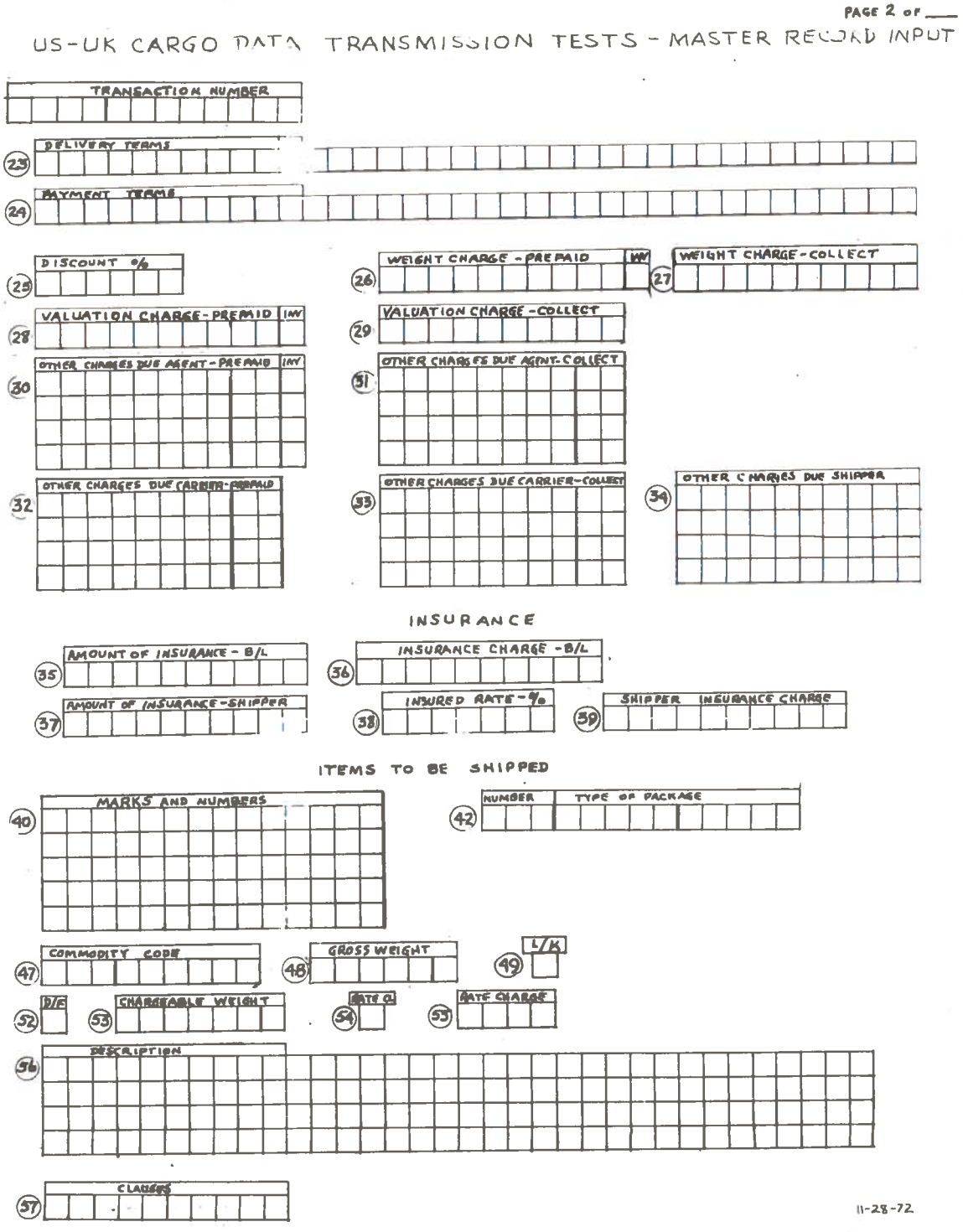

Figure 4. Data Input Form (Cont'd)

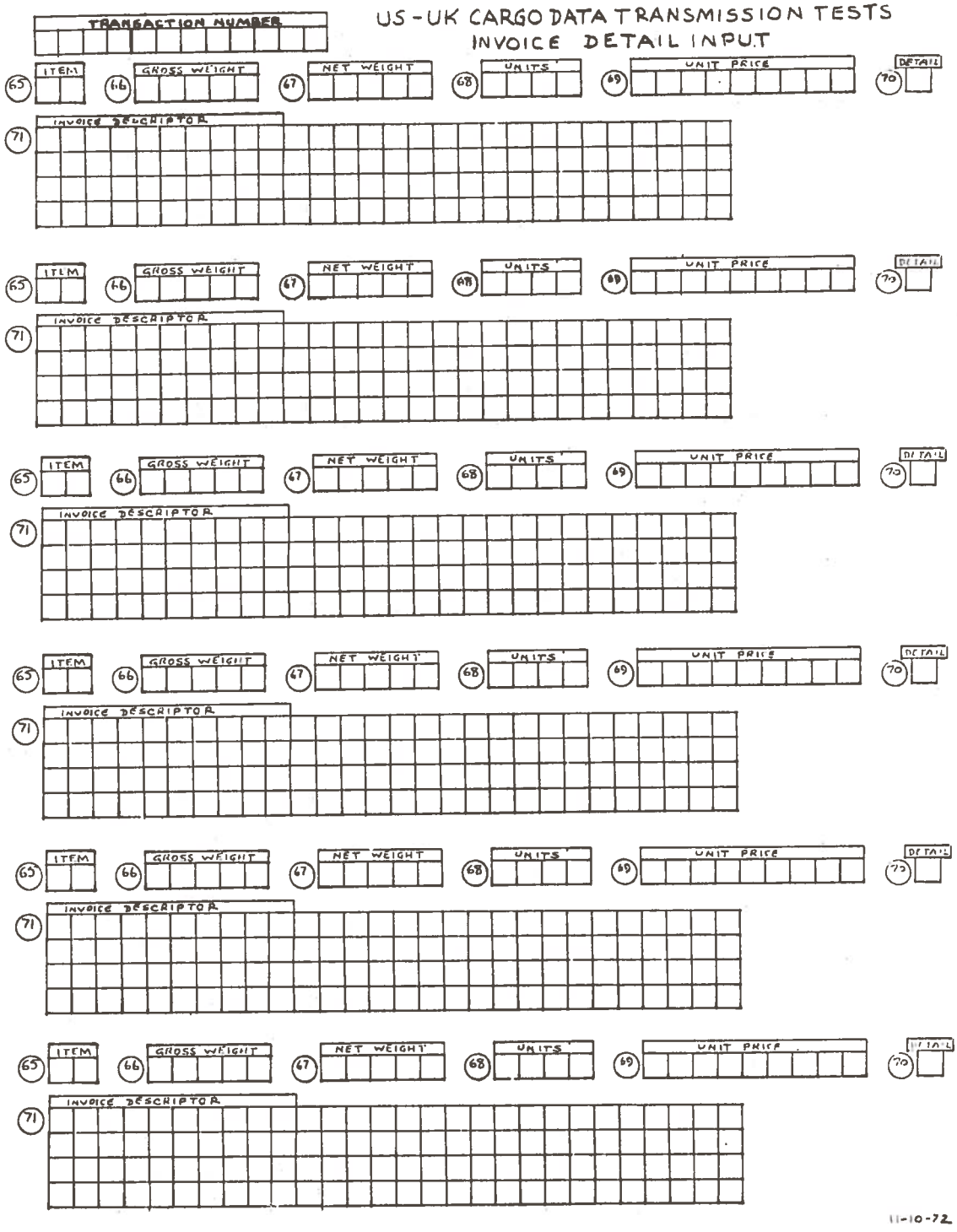

Figure 4. Data Input Form (Cont'd)

to that shipment are required. The program does not allow a given transaction number to be used for more than one shipment.

The system includes a querying program that allows data items for a shipment to be examined individually. To do so, the operator types the shipment number and the item number (using the field numeration on the input worksheet), and the computer types the data entered into the field. All addresses and commodity descriptions are typed out in full, even though they may have been entered in coded form.

An editing program is provided for adding, changing, or correcting individual data items for existing shipment files. The operator must type the transaction number and the item number and can then enter the data item. The same code look-up, format-checking, and error-correction features are available in the course of editing as during the original data entry. Any totals affected by a change in a component amount are recomputed automatically.

Data security is based on a system of passwords. To gain access to the system, the operator types a six-character password. If the password matches exactly one of a set of passwords stored in the computer, it identifies the operator as a particular one of the authorized system users. The operator can then perform all functions assigned to that user until the operator either gives the SIGNOF command or attempts access to unauthorized data. In those cases, no further access to the system is allowed until a valid password is typed in anew. The passwords, when typed, do not print, so that they can not be compromised if an unauthorized person sees the teletype paper.

The security system protects the data at several levels. No user can have access of any kind (for reading or writing) to a file containing data that is not his. Thus, shippers are restricted to access only data concerning their own shipments, and airlines only to those shipments assigned to them. In addition, each item of data is protected against unauthorized reading or modification. Table 1 lists all data items and identifies the parties with access to those data. It should be noted that there are many instances in

TABLE 1. LIST OF DATA FIELDS

FIELD NUMBER	CONTENT	PRINTED ON			ACCESS				
					Read			Write	
		Manifest	Bill of Lading	Commercial Invoice	Shipper/Agent	Airline	Government	Shipper/Agent	Airline
1.	Transaction Number	X	X	X	X	X	X	X	
2.	Flight Number and Date	X	X	X	X	X	X	X	X
3.	Departure Airport	X	X	X	X	X	X	X	
4.	Arrival Airport	X	X	X	X	X	X	X	X
5.	Origin Country		X	X	X		X	X	
6.	Shipper/Exporter		X	X	X	X	X	X	
7.	Consigned to		X	X	X	X	X	X	
8.	Intermediate Consignee		X		X	X	X	X	
9.	Notify Party		X	X	X	X	X	X	
10.	Send Invoice to			X	X			X	
11.	Deliver to Ultimate Consignee		X	X	X		X	X	
12.	Domestic Forewarder		X		X	X	X	X	
13.	Export References		X	X	X			X	
14.	Export License Data		X	X	X	X	X	X	
15.	General License Symbol		X	X	X	X	X	X	
16.	Country of Ultimate Destination		X	X	X	X	X	X	
17.	Customer's Order Number			X	X			X	
18.	Invoice Number			X	X			X	
19.	Invoice Date			X	X			X	
20.	Currency		X	X	X	X	X	X	
21.	Value at Point of Export				X		X	X	
22.	Declared Value for Carriage		X		X	X		X	
23.	Delivery Terms			X	X			X	
24.	Payment Terms			X	X			X	
25.	Discount			X	X			X	

TABLE 1. LIST OF DATA FIELDS (CONT'D)

FIELD NUMBER	CONTENT	PRINTED ON			ACCESS				
					Read			Write	
		Manifest	Bill of Lading	Commercial Invoice	Shipper/Agent	Airline	Government	Shipper/Agent	Airline
26.	Weight Charge - Prepaid		X	*	X	X		X	X
27.	Weight Charge - Collect		X		X	X		X	X
28.	Valuation Charge - Prepaid		X	*	X	X		X	X
29.	Valuation Charge - Collect		X		X	X		X	X
30.	Other Charges Due Agent - Prepaid		X	*	X	X		X	
31.	Other Charges Due Agent - Collect		X		X	X		X	
32.	Other Charges Due Carrier - Prepaid		X		X	X		X	X
33.	Other Charges Due Carrier - Collect		X		X	X		X	X
34.	Other Charges Due Shipper			X	X			X	
35.	Amount of Insurance - B/L		X		X	X		X	
36.	Insurance Charge - B/L		X		X	X		X	X
37.	Amount of Insurance - Shipper			X	X			X	
38.	Insured Rate			X	X			X	
39.	Shipper Insurance Charge			X	X			X	
40.	Marks and Numbers		X	X	X	X	X	X	
42.	Number and Type of Package	X	X	X	X	X	X	X	
47.	Commodity Code	X	X		X	X	X	X	
48.	Gross Weight		X		X	X	X	X	X
49.	L/K (Pounds/Kilos)		X	X	X	X	X	X	
52.	D/F (Domestic/Foreign)				X		X	X	
53.	Chargeable Weight		X		X	X		X	X
54.	Rate Class		X		X	X		X	X
55.	Rate Charge		X		X	X		X	X
56.	Description		X		X	X	X	X	

TABLE 1. LIST OF DATA FIELDS (CONT'D)

FIELD NUMBER	CONTENT	PRINTED ON			ACCESS				
					Read			Write	
		Manifest	Bill of Lading	Commercial Invoice	Shipper/Agent	Airline	Government	Shipper/Agent	Airline
57.	Clauses		*	*	X	X	X	X	
65.	Item		X	X			X		
66.	Gross Weight			X	X			X	
67.	Net Weight			X	X			X	
68.	Units			X	X			X	
69.	Unit Price			X	X			X	
70.	Detail				X			X	
71.	Invoice Descriptor			X	X			X	

*Print only when so indicated.

which a system user may read certain data fields but is prevented from modifying them. Bills of lading are available to all parties using the system (i.e. shippers, forwarders, airlines, and government offices). Invoices are available to shippers and forwarders only. Only the airlines can request manifests. The manifests are then typed on the teletype of the airline requesting them and also on a teletype representing government offices.

An operational cargo data management system would have provision for many simultaneous users. In the demonstration system implemented, only two simultaneous users are accommodated. One teletype may be used by users in any of the four categories - shipper, agent, airline, and government - and allows all functions of the system to be used. The other teletype has a more limited role. Its primary function is that of a government teletype. The functions allowed the government may be performed from this teletype in the same way as from the other. Thus, with the government password, one may obtain bills of lading and examine (i.e. - have the computer print) those data fields specified to be accessible to the government. This may be done concurrently with and completely independently of the operations performed on the other teletype.

There are two instances when the two teletypes are used together. Both occur when the general-purpose teletype is being used by an airline user. In both cases, the airline user assumes control over both teletypes. The first instance occurs when a manifest is being printed. The flight manifest is printed concurrently on the teletype serving the airline and also on the other (government) teletype. This demonstrates the feasibility of automatically filing documents with the government by having the system print them on a government printer. In this case the second teletype is still playing the role of a government terminal.

The other instance of non-independent use of the two teletypes has been implemented to demonstrate the usage of teletypes in a communications mode. After the command TALK given by the operator at the airline teletype, anything typed on the keyboard of either teletype is printed by both. This allows unformatted messages to be transmitted between them. The operator at the airline office

teletype can terminate this mode of operation by typing a $ sign.

A set of routines exists for producing five-level teletype paper tapes for transmitting invoices or bills of lading on five-level teletype networks such as the Western Union TELEX network. These routines produce the formatted documents from the data base constructed and validated using the system described, but they must be loaded and run separately. A seven-level teletype (ASR-33 or functional equivalent) is used to input commands to the system - i.e. to input document names and transaction numbers. The computer then types the documents on a five-level teletype (ASR-32) and punches a five-level paper tape. It automatically adds message headers, message terminations, and consecutive page numbers to make the tape convenient to transmit over a teletype communication network.

The entire system is implemented on a Honeywell DDP-516 computer with a core memory of 16,000 16-bit words, disk storage, and a dataphone interface capable of controlling three telephone lines. The program is written in DAP-16 assembly language and comprises about 8500 lines of source code. The computer does not have enough core storage to accommodate the whole program. Accordingly, the routines to accept input data and the routines to produce output documents are alternately assigned to the same block of core memory and are automatically loaded from disk when they are to be used. Even so, there are only a few hundred words of core storage left unused and available for expansion or refinement of the program.

The system as implemented permits one to demonstrate a number of the features and advantages that an operational central cargo data management system would have. In particular, the system allows the automatic production of several shipment documents with the minimum of manual effort. Any single item of data must be entered only once, even though it may be printed a number of times on the output documents. If documents are to be changed, only the particular items that must be changed need to be reentered. Data input is not only reduced but also simplified. The operator can enter data items one at a time in sequence without concern about precise

positioning on complex forms. Much of the data can be entered in concise coded form. The need to compute discounts and totals is eliminated. Certain kinds of errors are checked for and detected automatically, and all errors can be corrected without the need to repeat a whole form. Document dissemination is speeded by electronic transmission, and the various parties with access to the system obtain either entire documents or individual units of information on demand. At the same time, the data security system protects all information from any kind of unauthorized access.

The system as currently implemented can be expected to provide insights that should be useful in specifying features for an operational system. One might anticipate that various strengths and weaknesses of the system would become evident as attempts were made to use it in various contexts. Furthermore, exposure to the international trade community of such a rudimentary but working system should prompt useful suggestions for improvements and expansion from people familiar with trade documentation practices and requirements. It can be expected that contact with real equipment having some of the "feel" of an operational system would prompt practical suggestions that might complement the system specifications arrived at on a more abstract basis.

In demonstrating the system and in evaluating the responses to it, certain of its inherent limitations should be kept in mind. These limitations would not carry over to any practical operational system. They are largely simplifications dictated by the small size of the computer used and the generally restricted scope of the demonstration project.

The input-output capabilities of the system are very restricted relative to what an operational system would have. A practical system would accommodate not only many more simultaneous users, but also different types of input-output equipment, of which teletypes would be among the simplest and slowest.

The data structure in the current system assumes that every shipment needs only one commercial invoice (thus ignoring the possibility of consolidated shipments) and that every order is filled

with only one shipment (ignoring the possibility that one invoice might cover several separate shipments). The data files concerning a shipment are accessible only by reference to the air waybill number. No explicit record of changes to the data base is kept, no user is explicitly informed if another user alters a data entry that he has made, and no method is now provided to "freeze" all or some of the data at some particular stage. These are not serious limitations to a demonstration system that is conceived of primarily as a document generator. They clearly would not be acceptable in a system serving commerce and required to maintain strict accountability for information furnished and for large sums of money.

3. OPERATING PROCEDURES

3.1 MODE ACTIVATION PROCEDURE

The cargo data management program is designed to activate terminals connected to the DDP-516 Computer at the Transportation Systems Center in Cambridge, Massachusetts, through the telephone network. It is not a program running on a time-sharing system, so that specific prior arrangements must be made to reserve the computer for a given time and to have the program loaded.

The full capabilities of the system can be demonstrated with two terminals connected simultaneously. Either terminal may be disconnected and reconnected at any time without requiring any program restarts. The terminals must be the equivalents of ASR-33 teletypes with acoustic couplers. Other types of terminals may be used if their code signals are compatible (ASCII code at 100 baud - 10 characters per second). The formats typed out assume that output prints six lines per inch, at least 75 characters per line at ten characters per inch. The terminals must be operated in full duplex mode.

The program supports two terminals with somewhat different capabilities. The primary terminals accommodates users of all classes (shippers, forwarders, airlines, government), while the secondary terminal accommodates only "government" users. In addition, a mode of operation is possible in which communication is established between the two terminals; anything printed on one appears on both.

The primary terminal is connected to the computer by dialing (617) 494-2196 (an FTS number) and placing the telephone handset on the audio coupler when the connection is established and the carrier tone is heard. When the connection is first established, the terminal (which must be in line mode) will not give any response until a question mark (?) is typed on the keyboard. It will then print:

TYPE PASSWORD:

and the operator must type a string of six characters identifying the user to the system. These characters will not be printed as such, but the terminal will type a space for each one. If the six characters typed are a valid password, the computer will type:

OPERATION:

and accept instructions from the operator. If not (or if there has been a transmission error in communicating the characters typed to the computer), the computer will type:

TYPE PASSWORD:

and the operator must repeat the password.

The valid passwords are incorporated in the program and can not be changed by the user while the program is running. They can, however, be changed quite readily by simple modifications of the program. The set of passwords as defined on March 21, 1973 are listed in Table 2.

There are eight commands recognized by the system as valid operator responses to the system's request:

OPERATION:

These are listed and described briefly in Table 3 and described individually in detail below. In all cases, the computer expects a command of six characters, which will print as it is being typed. If the command consists of fewer than six characters (e.g. EDIT), the operator must type additional spaces or a single carriage return.

If the command typed is valid, i.e., one of the commands known to the system and authorized to the user, the computer will carry out the functions required. If the command is one that is known to the system but not legal for the user making it, the computer will type:

UNAUTHORIZED
TYPE PASSWORD:

and the user will have to type in a valid password, as above, before the system will accept any further commands. If the command given does not match any of the eight commands defined for the

TABLE 2. LIST OF SYSTEM PASSWORDS

PASSWORD	COMPANY	PASSWORD	COMPANY
LPANAM	PAN AMERICAN WORLD AIRWAYS	DOWCIN	DOW CHEMICAL CO. INTL. INC. INDIANAPOLIS, INDIANA
LLBOAC	BRITISH OVERSEAS AIRWAYS CORPORATION	DOWCMI	DOW CHEMICAL INTL. INC. (DEL) MIDLAND, MICH.
LLLTWA	TRANS WORLD AIRWAYS	GECINT	GENERAL ELECTRIC CO. INTL NEW YORK, N.Y.
GOVERN	GOVERNMENT	GECMED	GENERAL ELECTRIC CO. MILWAUKEE, WISC.
IBMWTC	IBM WORLD TRADE CORP.	GECWLM	GENERAL ELECTRIC CO. WEST LYNN, MASS.
DOWTEX	DOW CHEMICAL CO. LTD. FREEPORT TEXAS		

TABLE 3. LIST OF VALID OPERATIONS

COMMAND	OPERATION	PERMITTED		
		SHIPPER/AGENT	AIRLINE	GOVERNMENT
ENTER	Enter full set of shipment data	X		
EDIT	Modify data on file	X	X	
QUERY	Read individual data fields	X	X	X
WAYBIL	Print Bill of Lading	X	X	X
INVOIC	Print Commercial Invoice	X		
MANIF	Assemble and Print Air Manifest		X	
SIGNOF	Log off system, invalidate password	X	X	X

system, the computer types:

```
?
OPERATION:
```

and the operator may input a correct command.

If the operator notices a misspelling in an input command before the sixth character has been typed, the command can be corrected. To correct a character, the operator hits the RUBOUT key on the terminal keyboard and then types the character desired. The terminal will print the deleted character enclosed in backward slashes and then the character substituted.

The following example illustrates the method of correcting a single character. Assume the operator intended to type EDIT and has inadvertently typed EB. If the operator now depresses the RUBOUT key, the letter B will be deleted, and to indicate this the program will type\B, so that the total text will appear as:

```
OPERATION: EB\B
```

If the operator then types the proper character D, this will be accepted in place of the deleted B. The computer will type D, the \ serving to enclose the deleted character. The operator may then continue typing the command naturally. The completed line of typed test will appear as:

```
OPERATION:EB\B\DIT
```

This will be equivalent in all respects to:

```
OPERATION:EDIT
```

since the sequence of characters B\B\ merely denotes a B typed and then deleted.

If a character prior to the last one is found to be in error, it can be deleted by depressing the RUBOUT key several times. Assume, for instance, that the operator intended to type QUERY and did not notice an error until this stage:

```
OPERATION: QVER
```

The first depression of the RUBOUT key deletes the R and produces:

OPERATION: QVER\R.

The next RUBOUT deletes the E and produces:

OPERATION: QVER\RE

and the next RUBOUT finally deletes the V, leaving:

OPERATION: QVER\REV.

The operator can now type correctly "UERY" and the completed line will appear as:

OPERATION: QVER\REV\UERY

which is equivalent in effect to:

OPERATION: QUERY

since the sequence VER\REV\ merely indicates that VER was typed and then deleted once character at a time, starting from the right. By depressing RUBOUT, all characters including the first one of the command can be deleted. After that, depressing the RUBOUT key has no effect.

The operator should be forwarned about certain cases that may arise due to errors in transmitting the characters between the terminal and the computer. The terminal is operated in the "full duplex" mode. This means that there is no direct connection between the keyboard keys and the terminal printing mechanism. Instead, the depression of a keyboard key results only in a signal being sent to the computer. The computer sends back to the terminal the signals that produce printing. Errors are possible in the transmission in either direction, and will generally produce printing other than that expected. If an error in the printed text is detected which is attributed to a transmission error, it should be corrected in the same manner as an error due to human typing error. Thus, if the printed text appears as:

OPERATION: EB

even though the operator typed ED, the operator should press the RUBOUT key and retype the D. If the error occurred in transmitting the keyed character from the terminal to the computer, the response

to RUBOUT will be \B exactly as though the operator had mistyped. If the error occurred in transmission from the computer to the terminal, the printed line after the RUBOUT would look like:

OPERATION: EB\D

indicating that the character properly received and stored by the computer as D had been deleted. Not all signals distorted during transmission correspond to printing characters. Therefore, if no printing appears in response to a character key being depressed, the operator should not merely depress the key again, but should instead hit the RUBOUT key and then the character. Depending on when the transmission error occurred, the resulting line of printing might be either like:

OPERATION: E\\DIT (error in transmission from terminal to computer)

or

OPERATION:E\D\DIT (error in transmission from computer to terminal)

In theory, it is possible that multiple errors may occur in combinations giving rise to various confused printing patterns. In practice, transmission errors have been found to be quite rare. Badly garbled printing should be taken as an indication of either equipment malfunction or of a particularly noisy telephone connection, and diagnostic measures should be taken.

3.2 SYSTEM RESPONSE TO COMMANDS

This section describes the system response to the available commands. Throughout the parts typed by the operator are identified by underlining.

1) ENTER

The command ENTER is legal only to users identified as shippers or forwarders (i.e. using shipper or forwarder passwords). Its function is to create a data file for a shipment and to cause the program to accept the input data from the input work sheet, item by item, in numeric order. A detailed description of the input fields

is given in Section 4. Assuming that the user is one of those authorized to enter a case, the system response will be the following:

```
OPERATION:ENTER
INPUT
 1)
```

The operator must then type the data given on the input work sheet. Field 1 is the transaction (air waybill) number. If the operator attempts to enter in this mode data for a case that has been entered already, the system will not accept it. Instead, the following dialogue will result:

```
OPERATION:ENTER
INPUT
 1)TW--64244832

HAS BEEN INPUT ALREADY
OPERATION:
```

If the operator has entered a new transaction number, the program will establish a file for that transaction and automatically request the succeeding fields in numerical sequence. Insofar as possible, the program accepts input data in a format that matches the layout of the input data form (Figure 4).

In principle, the data fields can be grouped into some given types. In the first group are the fields like those within line 1 of the input form (Fields 1-5, 14-16, 17-21, 35-39, 47-55, 65-70). In entering data, the operator types the contents of each such field as given. For each field, the same error correction capability (via the RUBOUT key) is provided as described above for commands. All data are entered left-justified. A space has the effect of terminating input to the current field and causing the program to request the next field. A carriage return terminates input to the current field and causes the program to request the first field on the next line. Thus, the first line of input might look like

```
1)TW--64244832  2)_              3)JFK(CR)
6)
```

The underlined characters above are those typed by the operator. At the point shown above, the program would be waiting for input corresponding to data Field 6. The entry "TW--64244832" would have been stored as data for item 1, and JFK as data for item 3. No data would have been entered in Fields 2, 4, and 5. It should be noted that the carriage return (CR) can also be hit immediately after the computer has typed the field number to effect the skipping of the current field and all remaining ones on the line. Thus, for instance, the data called for in the last lines of the first page of the input form might be entered as follows:

14)_ 15)GDEST 16)GBR(CR)
17)(CR)
20)US 21) etc.

The effect would be no entry in Field 14, "GDEST" in Field 15, "GBR" in Field 16, no entry in any field on the next line (17-19), "US" in Field 20, etc.

In the second group of input fields are Fields 6 to 12. The data entered may be either in the form of a numeric code (which the program converts to an appropriate company name and address), or in the form of the full address. In addition, provision is made for entering the company address in coded form and augmenting it by additional text. The conventions for entering data for these fields are the following: if the input is to be in the form of full text, the operator simply types the text to be entered. Each line is ended with a carriage return. Two successive carriage returns signal the end of the field and cause the program to type the number of the next field to be entered. The RUBOUT key can be used to delete erroneous input one character at a time, as described in detail above. The RUBOUT key does not provide for corrections in any line preceeding the last. The back arrow (upper case O) key provides the capability of deleting the whole current line. Repeated use of the back arrow serves to delete succeeding lines one at a time, starting from the bottom. The back arrow character prints whenever a line is deleted. After the first (top) line of the field has been deleted, typing the back arrow has no effect - no further changes are made to the computer data base and no printing occurs.

If the input is to be a company address entered in coded form, the operator types only the numeric code, followed by a carriage return. The RUBOUT character may be used to correct typing or transmission errors. The program will reject any code not included in the table of codes for which equivalent addresses are stored in the computer and will request that the field be reentered, as follows:

```
6)123456789(CR)
NO SUCH CODE
6)
```

If the input is to be a company address entered in coded form, augmented by additional text, the operator must type the appropriate code number, followed by a space, followed by the text. The program will cause a carriage return and line feed to be executed in response to the space following the code number. The RUBOUT key can be used for error correction both while entering the code number and while entering the additional text.

It should be noted that the program distinguishes input intended to be a code number from input intended to be text by checking the first character. If the first character is a numeral, the program assumes that the input is to be code. If the input is a carriage return, the field is skipped. If the first character is anything else, all input to the field is treated as straight text. In either case, the first character can be deleted (by the RUBOUT key) and changed, but the interpretation that the input to the field is given by the program can not be changed once it has been determined.

The third group of input fields consists of Fields 13, 23, 24, 40, 56, and 71. All of these fields contain text in natural form. If the first character typed is a carriage return, the field is skipped. All other input is stored as it is entered. A single carriage return typed results in a new line; two successive carriage returns terminate the field. A RUBOUT serves to delete the last input character within a line for error correction. A back arrow deletes the whole last line.

The fourth group of input fields that are similar in format are Fields 30-34. All of these fields contain 0 to 4 lines of the general

form:

ABCD 1234

where ABCD represent any alphanumeric text and 1234 any right-justified dollar-and-cent amount with an assumed decimal point two places from the right (i.e., between the 2 and 3). At least one space must be present between the alphanumeric text and the dollar amount.

If the first character entered in a field of this type is a carriage return, the field will be skipped. Otherwise, data must be entered in the precise format for that line, and a carriage return must be typed at the end of the line. Two successive carriage returns cause input to that item to be terminated. The RUBOUT character may be used to effect error correction. The program will detect as errors and reject any input lines in which the carriage return is typed before the full ten characters for the line have been entered, or if a character other than a carriage return or a RUBOUT is typed after the tenth character.

2) EDIT

The EDIT command is used to modify individual fields of shipment data files that are already in the system. The command may be used by shippers, forwarders, and airlines. Representative dialogue involving the EDIT command would appear as follows:

OPERATION: EDIT(CR)
SHIPMENT NR: PA--62390370
FIELD NR: 17
17) 1234(CR)

FIELD NR: 5(CR) [or 5]
5) USA

FIELD NR: 0

OPERATION:

The following points should be noted: once the EDIT command has been given, field numbers can be specified in arbitrary order. The program will acknowledge the field number by typing it again and

accepting input to the field in the same form as it accepts it in the ENTER mode. A 0 (zero) given as a field number terminates the EDIT mode for the given shipment, and the program requests a new command. When the shipment number is entered, the RUBOUT key may be used to correct typing errors. When the field number consists of two digits, the program automatically gives a carriage return and line feed after the second digit and accepts the input data. When the shipment number is of only one digit, the operator must follow that digit with a space or a carriage return.

When the program expects the input to be a field number, it rejects as invalid any input character that is not a numeral or a blank. It indicates rejection by typing a question mark and a space following the improper character, and then awaits a proper field number (of 2 digits or one digit followed by a blank).

Thus, for instance, operator-computer interaction may look like:

OPERATION: EDIT(CR)
SHIPMENT NUMBER: PA--62390370(CR)
FIELD NR: 2A? 31
31)

Note that the program rejected both characters 2A because the A was non-numeric. The operator might have typed the A deliberately to delete the 2 originally typed in error.

Fields 65 to 71 are not uniquely described by field number alone, in that there is a separate entry so numbered for every item to be listed in the commercial invoice. When these fields are to be modified in the EDIT mode, the item number (1-10) must also be specified.

A sample of operator-computer dialogue leading up to the modification of one of these fields is given below:

OPERATION: EDIT(CR)
SHIPMENT NR. PA--62390370(CR)
FIELD NR: 66
ITEM 7(CR)

66.07)

Note that the program rejects non-numeric input in response to the request for ITEM numbers in the same way that it rejects such input in response to the request for FIELD numbers.

If a number is entered as a FIELD NR that does not correspond to any numbered field on the input form, an error message is generated, e.g.

```
FIELD NR: 50
NO SUCH FIELD
FIELD NR:
```

If an ITEM number is given outside the range 1 to 10, the program types a question mark and requests the item number again, e.g.

```
FIELD NR: 71
ITEM 12
?ITEM
```

If a user attempts to modify a field which he has no authority to alter, the program does not permit him to do so. For instance, should an airline user attempt to change the air waybill number of a shipment (i.e., Field 1) which only a shipper may modify, the attempt would be rejected as follows:

```
FIELD NR: 1
YOU ARE NOT AUTHORIZED TO DO THIS
FIELD NR:
```

A shipper, and only a shipper, may modify an air waybill number for one of his own shipments. This affords an opportunity to generate documentation for repetitive shipments by merely changing the waybill number and any variable items of a master file in which all the constant fields have been entered. It should be noted that in the EDIT mode, fields may be entered and changed in any order. The modified file is stored for subsequent retrieval only after the last change has been made and a 0 (zero) is typed in response to the request for a field number. This means that if Field 1, the air waybill number, is changed then all changes to other fields made after the same EDIT command will affect only the shipment data file with the new number, regardless of whether the modifications

to the other fields were made before or after the modification of the air waybill number. For instance, consider the following dialogue:

```
OPERATION: EDIT(CR)
SHIPMENT NUMBER: PA--62390370
FIELD NR: 17
 17) 04215(CR)
FIELD NR: 1
   1) TW--43219876
FIELD NR: 18
  18) 72824739
FIELD NR: 19
  19) 721217
FIELD NR: 0

OPERATION:
```

The effect of the editing has been to generate documentation for a shipment substantially like an earlier shipment. The customer order number, invoice number and invoice data (Fields 17 to 19) have been changed, and the shipment has been designated to go via TWA instead of PanAm. The data on file for the PanAm shipment have not been modified at all, even though the change to Field 17 was made before the new air waybill number was entered.

If an attempt is made while in the EDIT mode to create a file for an air waybill number for which a file already exists, the program produces an error message, aborts the EDIT procedure, and asks for a new operation.

3) QUERY

The QUERY command is used to get from the computer the contents of individual fields of shipment data files. Fields are designated by the numbers on the input form, and item numbers (1 to 10) are specified for the invoice items (Fields 65-71). When any one of the Fields 65 to 71 pertaining to an invoice item is asked for, the whole group for that field is typed out. As in the EDIT mode, the program rejects requests for non-existent fields or for data not

available for reading to the particular user requesting it. The following is a sample of dialogue involving the QUERY command.

```
OPERATION: QUERY
SHIPMENT NR: PA--62390370
FIELD NR: 7
CARRYFULL LTD.
LONDON AIRPORT NORTH
HOUNSLOW
MIDDLESEX ENGLAND

FIELD NR: 47

CHEMICALS
BTN 39.06 SITC 581.70

FIELD NR: 26
     104.83X

FIELD NR: 71
ITEM 1

A    208                          1                92.12

1 CTN ETHAFOAM WHITE
PLANK 9LB DENSITY 1.75 X
16x108/31228 64 000/PER CTN
19x19x112  IN

FIELD NR: 0
OPERATION:
```

Note that the fields can be asked for in any order, and that the full text equivalent is typed out for those fields (7 and 47) which have been entered in the form of code.

4) WAYBIL

The WAYBIL command causes the computer to type a bill of lading aligned with the U.S. Standard Master (Figure 1). A sample of the dialogue involving the WAYBIL command follows:

```
OPERATION: WAYBIL
SHIPMENT NR: PA--02668525
```

BILL OF LADING

DOW CHEMICAL CO. LTD PA--02668525
FREEPORT TEXAS
etc
OPERATION:

The RUBOUT command may be used in the conventional way to correct errors during entry of the command and the shipment number. No intervention by the operator is possible while the bill of lading is being typed, which takes approximately 1 1/2 minutes.

5) INVOIC

The INVOIC command serves to produce a commercial invoice aligned with the U.S. Standard Master format, with a continuation sheet where necessary. The conventions governing the use of the INVOIC command are the same as those for the WAYBIL command described above. Only shippers and agents are authorized to use this command.

6) MANIF

The MANIF command is available only to users identified as airlines. After the MANIF command is given, the program requests the operator to identify a flight by number and date and then to list by waybill number all shipments to be carried on that flight. The program automatically sets the flight number and date in the data file for each shipment thus assigned to a given flight to the number and date of that flight. After all shipments have been listed, the program types an air freight manifest on both the "main" teletype terminal used by the airline and on the "government" teletype. The manifests are in a format consistent with Customs Form 7509 (Figure 3).

Representative dialogue showing the use of the MANIF command follows:

OPERATION: MANIF(CR)
FLIGHT NUMBER: 123
DATE (DAY OF MONTH ONLY): 26
LIST SHIPMENTS

NR: BA--76743450(CR)
NR: BA--77058575(CR)
NR: BA--77058671(CR)
NR: BA--81825870(CR)
NR: BA--82282292(CR)
NR: (CR)
MANIFEST
etc.

The output that would then be produced on both the primary (airline) terminal on which the command and subsequent data were given and on the secondary (government) terminal is shown, with appropriate overlay, in Figure 3.

Note that the description of the goods in each shipment, as given on the bill of lading, is shown on the manifest. This is read from the appropriate shipment file. The month and year of the shipment and the origin and destination airports are fixed for the program. The ownership (here USA) is established by a table look-up procedure based on the airline.

The proper destination airport for all flights is assumed by this program to be London Heathrow Airport (LHR). The program examines the destination airport entry in the data file for each shipment whose number is given. If the arrival airport for the shipment has in fact been entered as LHR, the program proceeds normally. If the arrival airport given for the shipment is not LHR, the operator is given the option of either removing the shipment from the list to be included on the manifest or of including it and having the airport of arrival changed in the data file for the shipment. The date and flight number of a shipment are always automatically changed in the shipment data file to agree with the data and flight number of the manifest. The following computer-operator dialogue illustrates the exceptional cases that may arise in constructing a manifest.

OPERATION: MANIF(CR)
FLIGHT NUMBER: 123
DATE (DAY OF MONTH ONLY) 27
LIST SHIPMENTS:

NR: TW--76743203(CR)
DESTINATION AIRPORT GIVEN AS PAR
TYPE Y TO LOAD AND CHANGE DOCUMENTS
TYPE N TO DELETE FROM MANIFEST
N
OK, SHIPMENT REMOVED FROM MANIFEST
NR: TW--76748326(CR)
DESTINATION AIRPORT GIVEN AS LHP
TYPE Y TO LOAD AND CHANGE DOCUMENTS
TYPE N TO DELETE FROM MANIFEST
Y
OK, DESTINATION AIRPORT CHANGED TO LHR
NR: TW--76748327(CR)
NO RECORD
NR: PA--76748327(CR)
ILLEGAL
NR: (CR)
MANIFEST (etc.)

The response NO RECORD indicates that no data file exists for a shipment with the given number. The response ILLEGAL indicates that the shipment specified can not be included because it belongs to another airline. A special text NO CARGO is typed on the manifest in case no shipments at all are assigned to a given flight.

7) TALK

The TALK command establishes communication between the two terminals connected to the system. After the TALK command has been given, anything typed on either terminal will appear on both, with these exceptions: a carriage return on either terminal will result in both a carriage return and a line feed function on both terminals. A $ (dollar sign) typed on the secondary terminal will appear as $ on both. A $ (dollar sign) typed on the primary terminal serves to terminate operation in the TALK mode. When the $ character is typed on the primary terminal, the secondary terminal "goes dead" - i.e. will accept no input until a question mark is typed (whereupon it will ask for a password, etc.). The primary terminal requests a new operation command. A sample of operation in the TALK

mode follows, showing the typing on both terminals. The underlining indicates operator input at that particular terminal.

Primary Terminal	Secondary Terminal
OPERATION: TALK(CR)	
WHAT SHIPMENTS HAVE (CR)	WHAT SHIPMENTS HAVE
YOU GOT?	YOU GOT?(CR)
2 BOXES MARKED	2 BOXES MARKED
PA--62390415	PA--62390415 (CR)
ANYTHING ELSE?	ANYTHING ELSE?(CR)
NO OK$	NO OK
OPERATION:	

8) SIGNOF

The SIGNOF command terminates access to the system based on the current password. As soon as the SIGNOF command has been typed, the affected terminal appears "dead." It will not respond until a ? (question mark) is typed, whereupon it will ask for a password. The SIGNOF command would normally be given by an operator leaving the terminal in order to prevent any unauthorized individual from gaining access to the system.

3.3 PROGRAM LOADING PROCEDURES

The cargo data management program must be loaded in sections. The following paragraphs describe the procedures to be followed by the computer operators:

A. When the program is to be loaded for performing the various cargo data management functions using two eight-level teletypes, the following procedures must be followed:

1. Mount the CARGO DATA disk pack on the disk drive.
2. Set the Dataphones to the AUTO mode.
3. Load the TOP operating system.
4. Initialize to file directory 61.
5. Perform in sequence the following system commands:

TOP LC UPPER
TOP LC STATTA
TOP EX LOWER

The program will then be waiting for signals from a terminal connected to either of the dataphones, and will respond with a request for a password as soon as it has received a question mark.

B. When the program is to be terminated, put the MA-SI-RUN switch on SI, push MASTER CLEAR, and restart the computer at '17000 (the TOP operating system). If new data files have been created during the run which are to be saved, the following command should be given:

TOP SV STATTA
SAVE PARAMS 37000 37770(CR)

C. When the program for punching five-level paper tapes is to be loaded, the procedure is the following:

1. Set the dataphones to the AUTO mode.
2. Mount the CARGO DATA disk pack.
3. Load the TOP operating system.
4. Initialize to file directory 61.
5. Perform in sequence the following system commands:

TOP LC LOADII
TOP LC LOADIA
TOP LC DOWN
TOP LC BEARL
TOP LC BQLR
TOP LC LOAD3
TOP SW
*1000S

4. INPUT FIELDS AND FORMATS

This section details the various items of information included in the data base for each shipment, the input formats and conventions for these data, and the access rights to them. The items are listed in the order of their field numbers on the input form used for the demonstrations.

1. Transaction Number

The transaction number is the air waybill number. It consists of a total of 12 characters, in the format AA--NNNNNNNN, where AA are alphabetic and N's are numerals. The alphabetic part is a prefix designating the airline involved. The current program accepts only PA (=PANAM), TW(=TWA), and BA(=BOAC) as valid, and produces an error message "AIRLINE UNKNOWN" if an attempt is made to enter a transaction number with some other prefix.

When the transaction number has been entered, the list of shipment numbers for which data files exists is searched for that number. If the number is found to be in the list already, an error message is generated and the current ENTER or EDIT operation is aborted. If the number is not found, it is added to the list and a file is assigned to data for that shipment.

The transaction number may be entered by either shippers or agents. Only shippers may change it. The effect of changing a transaction number is to create a new shipment file which is a copy of the previous one with the transaction number charged. The old file is not deleted. (This allows multiple repeated shipments, using one partially filled-out file and adding the variable fields.) All system users may read the shipment number.

2. Flight Number and Date

The field consists of five characters, of which the first three are interpreted as the flight number and the last two as the day of the month. The month and year are supplied by the program. No check is made as to whether the date is valid. The data may be added or changed by the shipper, agent, or airline, and read by

all system users. The field is automatically made to agree with the flight specified in constructing a manifest if the given shipment is included in the manifest.

3. Departure Airport

These are alphabetic characters of the IATA airport code. These are converted to the full name of the airport for printing on documents. Only JFK (J.F. KENNEDY) is accepted as valid. All other airport designations generate a message "AIRPORT UNKNOWN." The field may be entered and changed by shippers and agents and read by all users.

4. Arrival Airport

The three-character IATA airport code is converted to full name for printing. All 3-character names are accepted, only LHR is converted (to HEATHROW). For all others, input characters are stored, to be examined by the MANIF program. The field may be entered by a shipper or agent, may be changed by the airline, and may be read by all users.

5. Origin Country

This is a three-character standard geographic code. The program accepts any three characters. The field may be entered or changed by a shipper or agent, read by government.

6. Shipper - Exporter

The code is a full address or Dun's Code of 9 digits, of which one is a check digit. Either may be followed by comments. The program takes the input to be the Dun's Code if the first character is a numeral. The list of available Dun's codes is then searched. If a match is found, the appropriate address is stored. If no match is found, an error message CODE NOT RECOGNIZED is typed out. The Dun's code must be given if the data file created is to be accessible to the shipper, since shipper password access is based on the Dun's Code. Access: entered or changed by shipper or agent, read by all users.

7. Consigned to

Same format, same access rights as for Field 6.

8. Intermediate Consignee

Same format, same access rights as for Field 6.

9. Notify Party

Same format, same access rights as for Field 6.

10. Send Invoice to

Same format as Field 6. Accessible only to shipper and agent for reading and writing.

11. Deliver to/Ultimate Consignee

Same format as Field 6. Access: for writing and reading - shipper and agent; for reading only - government.

12. Domestic Forwarder

Similar to Field 6, except that 7-digit IATA agents code is used instead of 9-digit Dun's code. The forwarder code must be given if the agent-forwarder is to have subsequent access to the data file. Access: as for Field 6.

13. Export References

Full text, not more than 3 lines of 36 characters (counting spaces) each. Accessible only to shipper and agent for reading and writing.

14. Export License Number and Expiration Date

The first 6 characters are interpreted as the export license number, the last 6 as the date, in the sequence year, month, day. In response to a query, the data in the field are typed out in the same form as they have been entered. However, on the bill of lading and the commercial invoice, they are expanded, so that for instance the entry V10495740731 is printed as:

EXP LIC NO. V10495 EXPIRES JULY 31, 1974

A validity check on the date is performed which requires that the month be indicated as a number between one and twelve, but no other validity check is performed on the input. Access: shipper or agent for writing or changing, all users for reading.

15. General License Symbol

Five characters of text, to be printed as they appear. The field carries the information whether or not the shipment is made under a general license. The valid entries to the field are either GDEST (in which case Field 14 would be blank) or no entry (in which case Field 14 would contain export license data). No checking is done by the program to insure that the entries to these fields are consistent, or that the entry in Field 15 is in fact GDEST. When export license information is printed on a bill of lading or a commercial invoice, the contents of Field 15 are printed unless Field 15 is empty. In that case, the specific license information (if any) in Field 14 is printed. The access rights for Field 15 are the same as those for Field 14.

16. Country of Ultimate Destination

Three characters printed as they are given. Access: for writing and changing - shipper and agent; for reading - all system users.

17. Customer Order Number

Any ten characters, which must be entered left-justified with no interspersed spaces. Accessible only to shipper and agent and for printing on commercial invoice.

18. Invoice Number

Similar in all respects to Field 17.

19. Invoice Date

Six characters, assumed to be pairs of numerals in the order year, month, day. Printed in conventional from MO/DA/YR. No validity check is performed. Access as for Field 18.

20. Currency

Three characters, printed as they are given. Accessible for writing and changing to shipper and agent, for reading to all system users.

21. Value at Point of Export

Up to ten characters, which must appear left-justified with no interspersed spaces. The proper entry to this field is the value in whole dollars. No validity check is performed. The entry is made for inclusion in export statistics reports only, not for printing on any shipping documents. Accessible to shipper and agent for writing and changing and reading, to government for reading.

22. Declared Value for Carriage

Up to ten characters, which normally should be left-justified dollars and cents. Any text is accepted. Accessible for writing to shipper and agent, for reading to shipper, agent, and airline. Printed on bill of lading (and thus available to government also - see Section 6.5, Known Weaknesses).

23. Delivery Terms

Any text of up to 36 characters, which must be terminated by two successive carriage returns. Accessible for writing and reading to shipper and agent.

24. Payment Terms

Similar to Field 23.

25. Discount Percentage

Input must be right-justified. The proper form of the input is either a blank or the percentage expressed in percent as a whole number, optionally followed by a decimal point and a decimal fraction. If a discount percentage is given, the amount of the discount is computed, and the discount percentage and amount are printed on the commercial invoice as well as being reflected in the invoice total. No check on the validity of the input format is performed by the program, and improper input can lead to garbled results. If the discount percentage is changed by editing, all amounts dependent upon it will be properly recomputed. The item is accessible for writing and reading to the shipper and agent only. It is printed on the commercial invoice if present.

26. Weight Charge - Prepaid

The field consists of two parts. The first is ten characters long, left-justified. It properly contains an amount expressed either in whole dollars without a decimal point, or in dollars and cents with a decimal point. A format validity check is not performed by the program. The second part of the field consists of one character to indicate whether or not the amount given is to be explicitly added as a charge on the commercial invoice. When the field is entered, a space typed in the first part results in input to the first part being terminated and the terminal typing several spaces. The operator may then type either a space or a carriage return, leaving the second part blank and indicating that the charge is not to appear on the commercial invoice, or any other character, which will be printed as an X and will indicate that the charge is to appear on the commercial invoice. The field is accessible for writing and reading to the shipper, agent, and airline. The amount as given is printed on the bill of lading and added to the prepaid freight charges. When indicated, the sum of the given amount and the prepaid valuation charge (Field 28) is printed on the commercial invoice and included in the invoice total.

27. Weight Charge Collect

A ten-character field which may properly contain a left-justified dollar amount (without a decimal point) or a dollar-and-cents amount (with decimal point). A format check is performed to assure that there are either no places or two places after the decimal point. The amount is printed on the bill of lading and properly added to the total collect charges. The field is accessible for writing and reading to the shipper, agent, and airline.

28. Valuation Charge - Prepaid

Similar in all respects to Field 27.

29. Valuation Charge - Collect

Similar in all respects to Field 26.

30. Other Charges Due Agent - Prepaid

A field of four lines of ten characters each. On each line, the proper entry is descriptive text (presumably some abbreviation of the nature of the charge) left-justified in the field, and a dollars-and-cents amount right-justified in the field. A decimal point is assumed, but must not be explicitly entered. It is added automatically by the program and appears when the item is printed. At least one blank must appear between the last character of the descriptive text and the first numeral of the amount. The program performs format-checking to insure that the amount is properly right-justified and requests that it be reentered if the input is not exactly ten characters per line. A carriage return typed at the end of a line allows the next line to be input if the current line is proper. Two successive carriage returns cause the remaining lines to be skipped. The field is accessible for writing and reading to shippers and agents, and only for reading to airlines. The information is printed on the bill of lading, and the charges are added to the total charges - prepaid on the bill of lading.

Note - The input form shows provision for an "INV" check mark after the first two lines of Field 30. The effect of a check mark in these boxes would be similar to that in Fields 26 and 28. This feature has not been implemented in the program.

31. Other Charges Due Agent - Collect

Similar in format and access rights to Field 30 as implemented.

32. Other Charges Due Carrier - Prepaid

Similar to Fields 30 and 31, except that the field contains only two lines, and airlines have access for writing as well as for reading.

33. Other Charges Due Carrier - Collect

Like Field 32.

34. Other Charges Due Shipper

Like Field 31 in format. Accessible for writing and reading to shippers and agents.

35. Amount of Insurance - B/L

A ten-character field which properly may contain a dollar-and-cent amount, to be input left-justified. No kind of format-checking is performed. The field is accessible for writing and reading to shippers and agents, for reading to airlines. The data are printed on the bill of lading.

36. Insurance Charge - B/L

Like Field 35 in format. Accessible for reading and writing to shippers, agents, and airlines. The amount is added to the total charges-collect, printed on the bill of lading.

37. Amount of Insurance - Shipper

Like Field 35 in format. Accessible for reading and writing to shippers and agents.

38. Insured Rate

A field of seven characters, which properly contains the percentage rate of insurance, left-justified. The program performs no check on the contents of the field. Accessible for reading and writing to shippers and agents.

39. Shipper Insurance Charge

Like Field 37 in format and access rights. The amount is printed on the commercial invoice and added to the invoice total.

40. Marks and Numbers

The field may contain arbitrary text. Properly there should be no more than five lines of 14 characters each, but the program during input does not prevent these limits from being exceeded. The field is accessible for writing and reading to shippers and agents, and accessible for reading to all system users.

41. Not used.

42. Number and Type of Package - This field is shown as a one-line field of 13 characters on the input form and appears as such to the operator during input. Internally, the field is considered to consist of three subfields. The first three characters are assumed to

be the number of packages. The next ten characters are divided into two subfields of five characters each. They are intended to contain a description of the type of packaging for the shipment, and are printed on the Bill of Lading and the Commercial Invoice vertically aligned one beneath the other. The field is accessible for writing to the shipper and agent, accessible for reading to all system users.

43-46. Not used

47. Commodity Code

The input is in the form of a nine-character alphanumeric code, which the program converts to two lines of text. The first line is a description of the commodity in English, the second line consists of the corresponding BTN and SITC codes. The complete list of codes currently included in the look-up table available to the program is given in Table 4. The program will reject any input which is not a proper code with a message "CODE UNKNOWN" and request that the input to the field be repeated. The printed output is always the text equivalent of the input code. The field is accessible for writing and reading to shippers and agents, and accessible for reading to all system users.

48. Gross Weight

A numeric field of six digits, to be entered left-justified. No validation is performed to assure that all the characters be numeric. When the field is entered in the EDIT mode, the program also automatically requests input for Field 49. The field is accessible for writing and reading to shippers, agents, and airlines, and may be read by the government also.

49. L/K

This field consists of one character only and designates whether all weights entered into the data file and printed on the shipping documents are given in pounds (L) or kilograms (K). Only L, K, or a blank are accepted as valid inputs. All others are rejected with an explanatory error message. This data item is stored

TABLE 4. COMMODITY CODES (ITEM 47)

3574112BB	Electronic Computer Parts BTN 84.55 SITC 714.92	2821142XX	Plastic Granules BTN 39.06 SITC 581.99
2818220XX	2.2 Dimethoxy Propane 98% BTN 29.00 SITC 512.09	2831190XX	Diagnostic Products BTN SITC 931.00
3679130XX	Electronic Components and Accessories BTN 85.18 SITC 729.95	3629215XX	Non-rotating Rectifier and NEC Parts BTN 85.01 SITC 722.10
3574112AA	Binary Dec. Converter BTN 85.01 SITC 722.10	2899991XX	Chemicals BTN 39.06 SITC 581.70
3071939XX	Plastic Liquid BTN 39.01 SITC 581.10	2821145XX	Synthetic Resin BTN 39.02 SITC 581.20
3629210XX	Electrical Equipment BTN 85.01 SITC 722.10	3551943XX	Diagnostic Equipment BTN 84.18B SITC 719.23
3612911XX	Transformers BTN 85.01 SITC 722.10		

in the shipment data file and printed in response to a QUERY command and on the shipping documents as "POUNDS" or "KILOS". The field is accessible for writing to shippers and agents, and to all users for reading.

50-51. Not used.

52. D/F

This field consists of one character, properly a D or an F, designating whether the goods are of domestic or foreign origin. No validity check is performed. The information is entered in the shipment file and may be read by a QUERY command, but is not printed on any of the shipping documents produced by the system. The field is accessible for writing to shippers and agents, and for reading to the government.

53. Chargeable Weight

A field of seven characters, which ordinarily should be numeric. No validity check is performed. The field is accessible for reading and writing to shippers, agents, and airlines.

54. Rate Class

A field of one character, which is not checked for validity. It is accessible for writing and reading to shippers, agents, and airlines.

55. Rate Charge

A field of four characters which normally contains a dollar-and-cent amount. The amount is not used in any calculation performed by the program, and no format validity check is performed. The access rights are the same as for Fields 53 and 54.

56. Description

This field contains text giving a description of the commodity being shipped to complement the description derived from the commodity code entered as Field 47. Any text is accepted. Proper format is not more than four lines of not more than 34 characters each. Access rights for this field are the same as for Field 47.

57. Clauses

This input field may contain up to ten characters, each of which may correspond to some standard clause to be printed on the bill of lading and/or commercial invoice. The letters used and the corresponding clauses are listed in Table 5. The clauses may be specified in arbitrary order. The program does not check at the time of input whether the input characters are valid - i.e., whether corresponding clauses have been specified. (Invalid characters merely result in no printing.) The field is printed in the form of the input in response to a QUERY command, but the full text of the clauses, without the code letters, is printed on the appropriate shipping documents. The field is accessible for writing to shippers and agents, and accessible to all system users for reading.

TABLE 5. CLAUSES (ITEM 57)

CLAUSE		PRINT ON	
		B/L	I
B	Safe for passenger aircraft	X	
C	Complete shipment	X	X
D	Partial shipment	X	X
E	Completing shipment	X	X
F	Delivery verification not required		X
H	A - emergency		X
J	Harmless chemicals	X	
K	CE emergency COP 3384		X
L	Red label material - restricted for carriage on cargo aircraft only	X	
M	Not restricted cargo	X	

58-64. Not used.

65-71. Description of Items for Commercial Invoice

These fields are repeated in groups up to ten times, with one group per item to be listed on the commercial invoice. The access rights to all of these fields are the same; they may be written and read only by shippers and agents. When these fields are to be input in the ENTER mode, the fields in successive groups are requested and entered in order. The program requests an entry not as e.g. "65)" but as "65.03)", where the ".03" indicates that the entry applies to the third item. The program repeats request for new items until all fields pertaining to an item have been skipped. The program then files the data for that shipment on the disk and requests a new OPERATION command. In the EDIT mode, any field for any item may be modified. The program requests the field number first. If the field number is between 65 and 71, it requests an ITEM number next and the operation may enter 1 to 10. The program then requests the specific data item designated in the same manner as in the ENTER mode. If any of the Fields 65-71 are requested in the QUERY mode, the program requests the ITEM number. The whole group of fields corresponding to that item number is then printed regardless of which specific field was requested. What follows are descriptions of these fields individually.

65. Item

Any two characters. It should be noted that the item designation in this field is not necessarily the same as the consecutive item numeration used by the program in the ENTER, EDIT, and QUERY modes.

66. Gross Weight

Five characters, left-justified. The weight must be given in whatever units have been specified in Field 48. No test is performed to check whether gross weight given is consistent with net weight given in Field 67.

67. Net Weight

Similar to Field 66.

68. Units

An integer of four digits or less, given left-justified.

69. Unit Price

Normally a dollars-and-cents amount, to be entered left-justified in nine characters or less. If the entry is an integer, it is assumed to be in whole dollars. The unit price is multiplied by the units and the product is printed on the commercial invoice and also used in calculating the invoice total. The unit price may be given to more than two places after the decimal point. The total amount will then be rounded to the nearest cent. If something other than a numeric quantity is entered in the unit price field, it is treated as zero in subsequent calculations. In such a case, nothing is printed in the TOTAL column on the commercial invoice (e.g., the entry for unit price might be N/C). A special provision is made in the current version of the program to accommodate participating shippers who wanted to explicitly conceal their prices. If the entry in any unit price field begins with the character X, then XXX.XX is printed for that item in the TOTAL column and also as the invoice total.

70. Detail

One character indicating whether the item described by the entries in the current group of fields is a major invoice item or merely a detail component of such an item listed previously. If a blank is entered in this field (or if the field is skipped by typing a carriage return at the end of Field 69), the item is taken to be a principal invoice item. If any other character is entered in the field, the item is taken to be a detail item. In response to a QUERY, the field is then printed as an X. The field as such does not appear on the commercial invoice, but items given as detail items are listed only on the continuation page of the commercial invoice. The current program assumes that if detail items are listed there will be only one major invoice item, given as the first item on the commercial invoice. The sum of the prices of all the detail items is printed on the commercial invoice as the price of that major invoice item.

71. Invoice Descriptor

Arbitrary text of not more than four lines of 28 characters each.

5. HARDWARE DESCRIPTION

The cargo data management system described here has been implemented on a Honeywell DDP-516 computer at the Transportation Systems Center (TSC) in Cambridge, Massachusetts. The DDP-516 computer is a machine generally considered to be one of the larger minicomputers. It has a word length of 16 bits and a cycle time of .96 microseconds. The particular machine used has 16,000 words of core memory, a moving-head disk drive providing approximately three million words of disk storage, and a number of other peripheral devices which are not used by the cargo data management program.

The system user communicates with the machine via a remote terminal connected to the computer through the telephone network. The remote terminals used have been ASR-33 teletypes. Any other code-compatible terminal can be substituted. If properly formatted output documents are to be produced, the terminals should print 6 lines to the inch (vertically), 10 characters to the inch (horizontally), and at least 75 columns.

During the tests and demonstrations of the system in early 1973, the transmission abroad of commercial invoices and air bills of lading was done by PanAm over the PanAm teletype network. The messages were transmitted to PanAm from TSC over the Western Union Telex network. This is a network that interconnects five-level (ASR-32) teletypes. The messages were prepared for transmission over the Telex network in the form of five-level paper tapes punched by an ASR-32 teletype driven by the computer through the telephone network.

The coupling of the computer to the telephone network is through a specifically constructed interface that controls three dataphone sets. The interface was designed and built by TSC. A duplicate is available as a spare. The functions of the interface are to act as a buffer and to perform the character format conversions between the bit-parallel format of the computer and the bit-serial format in which characters are transmitted on the telephone lines.

The interface to the computer appears as a "device" connected to the I/O bus and designated as device '56. A control word must be output to this device to establish how the succeeding input or output instructions to the interface are to be treated. The instruction formats for input-output instructions to the interface are, for output:

OTA 'N46

and for input:

INA '1N46

where N = 0, 1, or 2 and designates the particular data phone being addressed. The general format of the 16-bit control words is

1PSENN--------

where the first bit must be a one to designate a control word and the significance of the following bits is the following:

P=0 No parity bit
P=1 Parity bit
S=0 Two stop bits
S=1 One stop bit
E=0 Even parity
E=1 Odd parity

NN=00 8 bits/character
NN=01 7 bits/character
NN=10 6 bits/character
NN=11 5 bits/character

The eight low-order bits in the control word are immaterial.

When a character is to be output, the output word format is

0-------XXXXXXXX

where the first bit must be 0 to differentiate the output word from a control word, the next seven bits are irrelevant, and the final eight bits are the data (character).

An input word is in the format

-----LSPXXXXXXXX

where the first five characters are meaningless, the last eight are

the data (character), and L, S, and P indicate various error conditions, as follows:

L=0 Character OK, no characters lost
L=1 A character was lost (i.e., interface has received more than one character since the last time it was interrogated by the computer)
S=0 The stop bits received were OK
S=1 Bad stop bits
P=0 Parity consistent with that of control word
P=1 Parity error in character.

To alleviate timing problems in the interface circuitry, the unit number (N in the I/O instruction) must be preset by an OCP command before the I/O command is given. The instruction for doing this is

```
OCP 'N46.
```

The interface has provisions for operating at various different baud rates. Changing from one baud rate to another requires a plugboard wiring change.

For use with the cargo data system the interface was set to operate at 110 baud. Within the computer, all characters are represented as 8-bit ASCII codes. The characters received from the ASR-33 teletypes are in 6-bit ASCII code, and are converted by program. Output to the 5-level ASR-32 teletype is in 5-bit characters derived by table look-up. A different control word must be used. The following illustrative code segments show proper use of the interface for reading and writing on the ASR-33 teletype and for writing on the ASR-32 teletype.

Example 1 - To write one character - a question mark-on an ASR-33 teletype via dataphone 1.

```
OCP     '146            Select unit one
LDA     ='142000        Control word
OTA     '146
JMP     *-1             Loop until output successful
LDA     ='277           Question mark
```

```
OTA     '146
JMP     *-1                 Loop until output successful
```

Example 2 - To read one character from unit 2, check validity, convert to 8-level ASCII, and store in location X.

```
OCP     '246                Select unit two
LDA     ='142000            Control word
OTA     '246
JMP     *-1
INA     '246                Attempt to read
JMP     *-1                 Loop until successful
STA     TEMP                Temporary storage
ANA     ='3400              Blank out all but error bits
SZE                         Skip if no error condition
JMP     ERR                 Error handling routine
LDA     TEMP
ANA     ='77                Convert to
ERA     ='300               8-bit code
STA     X
```

Example 3 - To write a character on 5-level (ASR-32) teletype via unit 1.

```
OCP     '146
LDA     ='121000            Control Word
OTA     '146
JMP     *-1
LDA     CHAR
OTA     '146
JMP     *-1
```

Note that the control word is different, but the output procedures are essentially the same for the two different output teletypes.

6. PROGRAM DESCRIPTION

6.1 OVERVIEW OF THE PROGRAM

This section describes the cargo data management programs in some detail. The description is intended to give an overview of the entire system and of the various subroutines for the benefit of any programmer who might be called upon to change, correct, or augment the existing system, or possibly to implement certain of its functions on another system. This section is expected to be of little interest to non-programmers.

The cargo data management system described here is written in DAP assembly language to run on the DDP-516 computer under the TOP operating system. The program is loaded and execution is begun with TOP system commands entered via the on-line system teletype. Disk files are established, read, and written by calls to the TOP operating system subroutines.

The heart of the cargo data management programs is the input-output (I/O) routine. The I/O routine serves to link the user terminals (connected to the system through the telephone network) to the data manipulation routines carrying out the functions called for by operator commands. For each terminal connected to the system, there is in core a routine to service the terminal, i.e., to accept data, file them, retrieve them, manipulate them, and output them in appropriate form. Each of these routines consists of sequences of instructions that manipulate data in core, of calls to TOP subroutines that transfer data between disk and core, and of calls to the I/O routine to either accept input from the appropriate terminal or to produce output on that terminal. Whenever the I/O routine is entered from a particular data manipulation routine, it polls all (both) terminals connected to the system and attempts to perform whatever I/O operations are called for and are possible. After a character has successfully been transferred between the computer and a terminal, the I/O program checks whether all the data transfer called for by any of the data manipulation routines has been completed. Control is transferred to that data

manipulation routine whose I/O requirements have been satisfied, if there is one. This may or may not be the one from which the I/O routine was last entered. If the I/O requirements of no data manipulation routine have yet been satisfied in full, the I/O routine attempts to transfer further characters between the computer and the terminals until this is the case. The effect is that the I/O routine determines which of the data manipulation routines is carried out at any given instant. The success of the scheme depends on the fact that no data manipulation routine requires an excessive amount of time between calls to the I/O routine, so that all terminals are serviced frequently.

The data format for stored text is almost everywhere in the form of two 8-bit ASCII characters per word. A string of characters is preceeded by a single word containing the number of characters in the string. An arbitrary number of such strings may follow one upon the other. In the proper context they will be considered as successive lines of text in a paragraph. A negative number (-1 is used) in the character count location indicates an empty line, or the end of a paragraph.

Most output is produced by calling one of two subroutines (PLR for the primary terminal, RLR for the secondary terminal) which assume data in the form described above, reformat the data to be output into a form suitable for the polling I/O routine, and then call that.

Input is accepted by any of a number of subroutines, which typically call the polling I/O routine to read one character, store that character, call the I/O routine again to output the appropriate echo, which is generally the character just read (this is necessary since the terminals are operated in the full duplex mode) and then process the character.

The storage of data on the disk and retrieval of data from the disk are done by calls to the appropriate TOP operating system subroutines. The convention of the operating system require that the user identify each file by a six-character alphanumeric name (stored as three words). The cargo data

management programs store the data for each shipment on a separate disk file. The program maintains a table relating the shipment identification numbers to the corresponding TOP file names (FILE1 to FILE40 are used). When an operator enters a shipment number, the table is searched to establish whether a file for a shipment so identified already exists. The program may then read the existing file into a core buffer, create a new file, or produce an error message to the operator as required.

The cargo data management programs consist of some 8000 lines of code. With the TOP operating system and required storage buffers, they would require more than the available 16000 words of memory fit into core all at one time. Therefore, some routines are read into core only when required, using memory locations that at other times may be occupied by other routines. In particular, the routines to print out the bill of lading and commercial invoice on the primary terminal and to construct and print the air manifest use the same section of core as the input and edit routines. They are read into core in response to the corresponding operator commands from disk files OUTPUT and INPUTR respectively.

The program for a number of reasons was assembled in parts, and has to be loaded in sections. In the first place, the TOP operating program occupies sectors 13-17, so that the core area available for user programs is not contiguous. The system requires that any program to be loaded go into a single contiguous area of core. To meet this requirement, the parts of the program occupying the lower-numbered section had to be assembled separately from those occupying the higher-numbered sections. Furthermore, it was found that the program section intended to occupy the low-numbered sections had too many variable names for the assembler to handle, and therefore had to be broken into parts for assembly. A sector-by-sector description of the contents of core is given in Table 6.

The table relating shipment numbers to corresponding file names is maintained in sector 37. If new files have been added during a program run and it is intended that they be kept in the data base, this table must be stored at the conclusion of a program

TABLE 6. CORE STORAGE UTILIZATION

Cargo Data Management Program	
Sector	Contents
0	0-'377 Bootstrap loader, TOP links '400-'777 Program links
1	Master program for primary terminal EDIT, QUERY routines
2	Primary teletype subroutines, utility subroutines
3	Text and code reading subroutines for primary terminal (for ENTER and EDIT function)
4	Primary teletype line printing routine (PLR), utility subroutines
5	Polling routine
6-12	Region for reading in alternative subroutines sets: a) From file INPUTR Routines to accept every input field under either ENTER or EDIT commands b) From file OUTPUT Routines to print output documents on primary teletype (under teletype WAYBIL, INVOIC, and MANIF commands)
13-17	TOP operating system
20-25	Buffer for shipment data files used for EDIT, ENTER, QUERY, WAYBILL and INVOIC commands on primary terminal.
26	Utility subroutines
27-31	Buffer for reduced shipment data files used for QUERY and WAYBIL commands on secondary terminal
32	Master program for secondary terminal, secondary teletype line printing routine (RLR)
33-34	WAYBIL routine for secondary terminal
35	Temporary storage used by subroutines
36	Code look-up tables
37	File directory relating shipment numbers to disk file names.

run. It must be separately loaded whenever the program is to be executed. Currently, disk file STATTA is used for this table. If new files have been created during a run, the TOP command to save a new version of the table should be given as follows:

```
TOP SV STATTA
SAVE PARAMETERS 37000  37775 (CR)
TOP
```

Clearly any other available file name may be used for this purpose, if care is taken to reload the proper file when the program is again to be executed.

The program at various places accepts input in coded form and converts the codes to equivalent text. In particular, a list of DUNS codes is used to identify various commercial concerns which are involved in the shipments, a list of IATA agents' codes is used to identify forewarders, and a list of commodity codes is used to describe goods on the bills of lading. The lists of codes (stored in alphanumeric form, 2 characters per word) are maintained in sector 36, along with pointers indicative of the location of the corresponding text in disk file DOWN. When the text is to be retrieved, file DOWN is opened and the contents are read in successive small increments until the proper location is reached in the file. The required segment is then read and transferred as needed, and file DOWN is closed.

The succeeding sections describe some of the key subroutines in finer detail.

6.2 OUTPUT ROUTINES

6.2.1 Polling Routine

The polling routine controls input or output on four devices simultaneously. It has eight entry points, one each for input and output on each of the four devices, and is called as follows:

```
JST     ADDR
DEC     N
DAC     LOC
```

where ADDR is one of the eight entry addresses, N is the number of characters for input or output and LOC is the location of the character string to be output or the location where input will be stored. The character format is one character per word right justified.

When the polling routine is entered for either input or output, it will check each of the four devices in sequence and attempt to input or output one character when input or output is pending. When input or output is successful the number of characters remaining for that device is reduced by one. When input or output is not successful the routine will not loop back and try that character again, but will go on to try the next device. This prevents delays on one device from affecting the operating speed of the other devices.

After one check of each device the routine looks to see if an input or output string has just been completed (device still in input or output mode and character count has just become 0) and if so, an exit is made from the routine to the address set by the JST for that input or output string. The checking for a completed string always begins one device past the latest entry device so that a series of one character calls will not result in the routine never getting to an exit at a following device. If no exit is found, the routine loops back to try another input or output character on each device.

The following variables keep track of the status of each device:

TN1 - 4. The number of characters remaining in an input or output string.

TM1 - 4. The status of each device. +1 indicates input mode; -1 indicates output mode; 0 indicates neither input nor output.

TP1 - 4. Points to the location of the next character to be output or the location where the next input character will be stored.

The current program has 8-level teletypes for devices 2 and 4, and a 5-level teletype for device 3. Device 1 is currently unused.

6.2.2 Setup Routine

At the start of the main program the setup routine should be called. This routine sets exit addresses in the polling routine for each device and sets the number of characters to zero and the mode to input mode for each device. The result is that after one character of input or output from the first call to the polling routine the routine will exit to a section of the program using another device; after the next character it will exit to a third section and after the next it will exit to a fourth section. If there is no section of the program for one of the devices, zero should be entered as the exit address and the routine will store zero as the mode for that device so no exit will occur.

The routine is called as follows:

```
JST   TSET
DAC   EX1
DAC   EX2
DAC   EX3
DAC   EX4
```

where EX1 to 4 are the addresses of the four sections of the main program using each of the four input-output devices.

6.2.3 Line Printing Routine

An item to be printed, such as an address, shipment number or special message, is stored in the computer memory as follows:

```
LOCA  DEC  N1
      BCI  N1/2, TEXT OF LINE 1
      DEC  N2
      BC1  N2/2, TEXT OF LINE 2

      .....
      .....
      DEC  -1
```

where LOCA is the location of the item; N1, N2, etc. are the numbers of characters in lines 1, 2, etc.; and the last line is followed by -1.

Since a line of printing may contain several items beginning in different columns, the call to the line printing routine is as follows:

```
JST   PLR
DEC   NL
DEC   NI
DAC   LOC1
DEC   COL1
DEC   LIN1
DAC   LOC2
DEC   COL2
DEC   LIN2
.........
.........
.........
```

where NL is the number of lines to be printed and NI is the number of items involved. Each item is described by three parameters; the location of the text in memory (LOC1,2, etc.), the column in the line where it begins (COL1,2,etc.), and the line within the item where printing begins (LIN1,2,etc.).

To print a document, one call to the printing routine is needed when each item originally appears, with the exception that if more than one item originally appears on one line, only one call is necessary.

Blank lines result when the number of lines specified exceeds the lines in the text. This system obviates assembling all of the text for a document in a buffer area and thereby saves time and memory space.

When the line printing routine is entered, it first stores the parameters as index counters and pointers in the proper format for use within the routine. It then moves the pointer for each

item to the beginning of the line within that item at which printing begins for the first line, making sure not to go past the end of an item.

The next step is to set up a line of output. Pointers are set for transferring the text to an output buffer, and counters are set for the columns and the number of items. If blanks are needed before the text for the first item, they are laoded starting with column one; then the current line of the first item is stored, then more blanks if needed up to the starting column of the next item, then the text for the next item, etc. One word of text containing two characters is stored in the output buffer as two words of one character each. The pointers and counters are incremented after each pair of characters is stored. After each item has been stored, the routine checks for line feeds or blanks starting at the current point in the line buffer and removes them, working backwards, until text is encountered or until the column pointer is back to column one. This is done so that if a line of text in one item runs past the starting column for the next item, no harm will be done if the overflow characters are blanks or line feeds. Also, this procedure removes the unwanted blank or line feed that occurs when an odd number of characters is stored in the buffer. Most importantly, it saves time during output since the unnecessary blanks are not typed.

When a complete line of output has been set up in the output buffer, three calls are made to the polling routine. The first call does a carriage return, the second does a line feed and the third types the output line. When a line is empty, only the carriage return and line feed are done.

After outputting a line, the routine loops back and sets up the next line, etc., until the required number of lines, including blanks, have been done.

6.2.4 Bill of Lading and Invoice

The bill of lading and invoice are printed by a series of calls to the line printing routine. Some of the printing involves

special circumstances. For example, the number of lines printed by one call may depend on how many lines have been required elsewhere; if a certain field is empty, another may be printed in its place; a message may or may not be printed, depending on the contents of another field. Also, on the invoice, a continuation page may or may not be needed. In order to check whether a field is empty or not, a subroutine called CHEK is used. The following is a description of CHEK and the construction of the bill of lading and the invoice.

6.2.4.1 Subroutine CHEK - CHEK is called as follows:

```
LDA   I$F
JST*  CHA
```

where I$F is the location of the field to be tested and CHA is the address of CHEK.

A field is considered to be empty if its first word contains a negative number or if the first word of the first line contains a pair of blanks ('120240). If CHEK finds the field to be empty, it returns to the next instruction. If the field is not empty, CHEK skips one instruction when it returns.

6.2.4.2 Bill of Lading - Since there are variable numbers of lines in field I$40 (Marks and Numbers) and the clauses determined by field I$57, the number of blank lines after the clauses must be variable so that the following lines start in the right place. The number of these blank lines is originally set to ten.

A check is made of field I$7 (Consigned to). If it is not empty it is printed. If it is empty, Field I$11 (Deliver to/ Ultimate Consignee) is printed in its place.

Field I$8 (Intermediate consignee) is checked. If it is not empty it is printed preceeded by "INTERMEDIATE CONSIGNEE".

A jump to LCN= gets the line count for field I$40. If this count is zero it is set to one. This count is used to set the number of lines to be printed for field I$40 and to reduce correspondingly the number of blank lines after the clauses.

Field I$15 (General License Symbol) is checked. If it is not empty it will be printed "GDEST". If it is empty, the following license information is printed: "EXP LIC NO.", Field I$14, "EXPIRES", Field I%14 and Field I#14. These fields contain the license number and expiration date, respectively.

Field I#57 contains ten words which determine whether special clauses are printed or not. Each word is checked in sequence, and if it contains a B,C,D,E,J,L, or M, the corresponding clause is printed and the number of blank lines following is reduced by the number of lines in the clause.

6.2.4.3 Invoice - A major difference between the construction of bill of lading routine and the invoice routine is that the invoice routine may require a continuation page. The following set of items is printed by one call to the printing routine: I$65 (Item), I$66 (Gross Weight), I$67 (Net Weight), I$68 (Units), I$69 (Unit Price), I$71 (Invoice Descriptor), and I$83 (Total Price). Up to ten of these sets may be printed on one invoice. Up to ten entries may be stored in memory, therefore, for each item. These entries are stored consecutively with a fixed number of words each so that a pointer may be aimed at the first entry and then advanced a fixed number of words so that it points at the second entry, etc. The entry lengths are as follows: three words for each I$65, five words for each I$66, five words for each I$67, four words for each I$68, seven words for each I$69, sixty words for each I$71, and seven words for each I$83. The number of lines required to print one set is obtained by finding the number of lines in I$71. There are only fourteen lines available for printing these sets on the first page of the invoice, minus whatever number of lines is required for printing the clauses generated by I#57, minus two lines for printing Field I$25 (Discount) if I$25 is not empty, and

minus one line for printing Field I$49 (Pounds/Kilo) if I$49 is not empty. Therefore, before a set is printed, the number of lines it requires must be compared to the number of available lines remaining. If not enough available lines remain, the set and all following sets are printed on the continuation page. A set may be deferred for printing on the continuation page for another reason. There is a Field I#70 (Detail) which contains ten words corresponding to the ten possible sets for printing. If the contents of the word in I#70 corresponding to a given set is not zero, that set is printed on the continuation page. In this case, however, subsequent sets may still be printed on the first page. A ten word "print array" is used to keep track of which sets have been printed. A one means the set has been printed, a zero means it has not been printed. Any available lines not used on the first page are printed as blanks, so that following items start on the proper line.

Other special considerations involved in the invoice routine are as follows:

Field I$18 (Invoice Number) is checked. If it is not empty, it is printed, preceeded by "INV. NO."

A check is made of Field I$7. If it is not empty it is printed. If it is empty, Field I$11 is printed in its place.

Field I$17 (Customer's Order Number) is checked. If it is not empty it is printed, preceeded by "CUST ORD NO."

If Field I$10 (Send Invoices to) is not empty, it is printed, preceeded by "SEND INVOICE TO".

The printing of clauses resulting from a scan of field I#47 is the same as on the bill of lading, with the exceptions that the letters looked for are A,C,D,E,F,H, and K, and that the scan must be done twice-once to find the number of lines that will be required in order to reduce the number of available lines accordingly and once to print the clauses.

6.2.5 Manifest

The manifest routine prints data concerning several shipments using calls to the line printing routine. The data for each shipment must be read from disk.

The manifest routine is called as follows:

```
JST   MFST
DAC   MFSA
DEC   N
```

where MFSA is the table address for the shipments to be printed. The entries in this table are used (see below) to find the file name addresses for each shipment. Entries are seven words apart. N is the number of shipments.

The routine first prints some lines that are independent of individual shipment data. It then stores the shipment table address in the "file name fetch" routine and in the "move shipment number" routine. The routine complements the number of shipments and stores the results (unless zero, in which case an error message and exit results) in a counter.

The routine then goes through the following loop for each shipment:

Using a jump to MST= , it gets the shipment's file name address and stores it in the "open file" routine.

Open the file on unit 2 for reading.

Loop to read through unneeded words at the beginning of the file.

Read in 110 words of Fields I$47 (Commodity Code) and I$42 (Number and Type of Packages) into a buffer.

Close the file on unit 2.

Move the shipment number to the print area by a jump to MRE=.

Add 7 to the address table pointers to be ready for the next shipment.

Print the shipment data.

At the end of the loop, print 52 blank lines (ejects rest of page) and exit.

6.3 INPUT PROGRAMS

All input of data to the cargo data files is performed in response to either the ENTER or the EDIT command, and is performed only from the primary terminal. Essentially the same code is utilized by both commands. These programs store in appropriate locations in core memory all information to be printed by the various output programs. Each field is stored in the format of text, two ASCII characters per word. Each string of words representing a line is preceeded by a single word containing the number of characters in the line. The character count in the next line of multi-line field is assumed to be the word immediately following the last word of text. When no more text is to follow, the character count is minus one. In addition to text, the numerical values of some data fields are stored in memory at the time they are input. When all input in response to an ENTER or EDIT command has been completed, the program computes the various totals to be printed on the bill of lading and the commercial invoice, converts the numerical values to the form of text, and then stores the contents of the contiguous core locations containing these data as a file on the disk.

Each field on the input form is processed by a subroutine which involves other subroutines that do the specific functions of reading data from the teletype, manipulating them as necessary, and storing them. When operation is in the EDIT mode, a master routine requests the number of the fields to be entered and calls the subroutine to read that field. Each such subroutine returns to this master calling routine, which repeats the operation. When operation is in the ENTER mode, each such subroutine arranges to exit to the next one to be used. Variable INIE is set to one when the program is in the ENTER mode and to zero when the program is in the EDIT mode, and each subroutine examines this variable to determine how it should exit. On the input form, there

are a number of cases in which several input fields are grouped on one line. The input convention requires that if a carriage return is typed in the course of entering such a line, the rest of the fields be skipped. The field entry subroutines for the fields involved meet this requirement. When the program is in the ENTER mode, the exit determined by each affected subroutine is set to the next successive input field if no carriage return is typed, and to the first field in the next line if a carriage return is typed.

The field input subroutines as a class all perform the following functions, invoking other subroutines to perform them:

a. Check the access rights to the field. Each password has associated with it a word specifying field access rights. When a password is given on the primary teletype, this word is stored in memory location INPW. Each of the first twelve bits in this word specifies an access right. The first three are assigned to shippers, the next three to agents, the following three to airlines, and the following three to the government. Within each group, the first bit designates the right to write (ENTER), the second the right to change (EDIT), and the third the right to read. If the bit is zero, the right is denied; if it is one, it is given. As currently implemented, every user password in each class (shipper, airline, government) has associated with it all the rights, insofar as users of that class have them. However, this structure permits, for instance, that a given airline have several passwords, of which some would be available to such employees as might need access to shipping data, but would have no right to change any stored data, and others which would give full access rights. Each field input subroutine matches a word specifying access rights to that field against the access rights word stored in INPW, and allows input only when the access right is confirmed. For instance, if an airline password has been given, the third set of three bits (and only these bits) of INPW are

set to be ones. Field 40 (Marks and Numbers) may be written and changed only by shippers and agents. The subroutine for accepting this field therefore checks INPW against an access word whose first, second, fourth and fifth bits are ones, and the rest zeros. None of the ones in one word match the ones in the other, and the subroutine would therefore print an error message and request a new field number for input.

b. Print the field number. A subroutine is called which prints the field number followed by a right parenthesis. Routine CLS (with indirect address INLP) prints several spaces, followed by two characters specified in the call, followed by a right parenthesis. Routine CWL (with indirect address INPR) first performs a carriage return and line feed.

c. Read the data into a core buffer location. Depending on the format of the data to be accepted, a subroutine is called for reading the input from the terminal into a core buffer. For those fields for which the proper input is either a code or the equivalent text, a rather complex routine CTR accepts the input and stores it in the proper location. For those fields in which the proper input is text that may consist of several lines, the input routine is DIIR. Both these routines read the text, accept corrections using the RUBOUT character, and format the text in standard format. For those fields which contain only one line, the standard input routine is WLI, which accepts input and stores it in a buffer WBUF, one character per word. WLI accepts corrections made using the RUBOUT key. Input is terminated either when a carriage return is typed, or when the number of characters to be accepted has been reached, or, optionally, when a space is typed. In such a case, the program tabs foreward (i.e., types out the number of blanks that would be required to fill the field). The right-hand part of the field to which no explicit

input has been given is always filled with blanks. In returning to the calling program, WLI skips the location immediately following the calling parameters, unless input was terminated by a carriage return. This provides the opportunity to skip fields as required by the conventions of the ENTER mode.

d. Perform any data checking, data conversion, or table look-up required. The specific routines are annotated in the program listings.

e. Transfer the data from any temporary input buffer to the memory locations from which they are to be filed on disk. Two different subroutines are generally used for this purpose. PAK takes a string of characters stored one per word in consecutive core locations and puts them into the standard format of two characters per word, preceeded by a character count and followed by a minus one. MRR moves strings of consecutive words, up to and including a terminating minus one from one core location to another.

6.4 DATA FILING

The data for any one shipment are filed on the disk as one file of some 3000 words. The file is created, read, and written by routines within the TOP operating system, which is responsible for the selection of physical disk record addresses and the various essential bookkeeping functions. The TOP operating system expects user programs to refer to files by names six alphanumeric characters in length. The files are known to the cargo data program user only by the air waybill (bill of lading) numbers of the associated shipments.

Sector 37 of memory contains a directory associating the air waybill numbers with a set of 6-character file names (FILE1 to FILE40). When a command is given that requires that a shipment data file be read from the disk, the list of air waybill numbers for which files have been created is searched. If

the particular number is not found, an error message is typed to the operator and the program requests a new operation command. If the waybill number is found, the associated TOP file name is determined, and that file is read into a block of core. If the command comes from the primary teletype, the whole file is read in. If the command comes from the secondary teletype, only part of the file (without the part pertaining to the commercial invoice) is read into core memory.

Access rights for the file are established before the user is allowed to read any of the data. The program first establishes the class of the user from the access rights word associated with the password. Government users have a right to parts of all files. Airline users' access depends on whether the prefix of the air waybill number applies to their airline. If it does not, access is denied immediately and no attempt is made to read the file into core. No agents have been assigned passwords in the current system; therefore, it has not been necessary to implement a check on an agent's right to a file. Such a check would be similar to that performed for checking shippers' access rights. If an operator requesting a file for any purpose is determined on the basis of his password to be a shipper, the file is read into core memory. Within the file there is stored the DUNS code for the shipper for that shipment. If that shipper is found to be the same one as the shipper requesting access to the file, he is given that access. Otherwise, he is barred from further access to the file.

If a file has been read into core for printing a document or in response to a QUERY command, it is not modified and therefore does not need to be written back onto the disk. When a file is to be modified (in response to an EDIT command or in the course of constructing a manifest), the whole file is read into core, those fields that are to be changed are stored in core in the appropriate places, and the file is eventually stored back on disk in the same place.

New files can be established in only two ways. The more common is in response to an ENTER command. The memory locations in which the file is to be assembled are cleared (loaded with zeros where numeric values are stored, and with minus ones where

text is stored), and a TOP file name is assigned to the shipment when the air waybill number (Field 1) is entered. If the air waybill number already has a file assigned to it, an error message is produced, the operation is terminated, and a new operation command is requested from the operator. No file is established. It is also possible to create a new file while operating in the EDIT mode by entering a new shipment number in Field 1. The program then checks whether the waybill number entered is a duplication, and assigns a file to the shipment if it is not. If the number entered is a duplication, the program leaves the EDIT mode with an error message. Any changes made to any fields of the original file before the new shipment number was entered are lost. The original file is unaffected.

6.5 KNOWN WEAKNESSES

This section lists known weaknesses in the software that may lead to difficulties when the program is run. For the most part, these weaknesses are of the nature of failures to consider some particular type of input which normally should not occur and for which no proper response has been explicitly specified. These flaws have not been removed from the present system because they would manifest themselves only in response to inputs that were not proper in any case in the context of the demonstrations to be conducted. Because there is always a danger in programs as complex as this one that corrections made for some known condition can introduce errors for other conditions that may be very difficult to diagnose when they are eventually discovered, and because the total time and effort available for making the program operational were limited, the decision was made to "freeze" the program as it stood. In a program to be run under fairly tightly controlled circumstances, the known "bugs" are acceptable since there is no valid reason for making inputs that would cause them to appear. In an operational system, bugs of this type are definitely not acceptable, since the program must be made immune to any improper but physically possible input lest it be disabled due to operator ignorance, accidents, or even deliberate sabotage.

Some of the weaknesses are of the nature of omitted features, which an operational system would need but which have not been implemented on the present system. The possible problem areas are the following.

6.5.1 File Directory Overflow/Disk Capacity Overflow

If in the course of entering data, an attempt is made to create a file after the TOP file directory contains 100 files already, or an attempt is made to write a file after the disk is already full, a TOP operating system error message will be typed on the computer console teletype and the operating system will assume control. The remote terminals available to the users will "go dead." This kind of system crash can be prevented if sufficiently large dummy files (File 1 to File 40) are created artificially ahead of time. That may well be inconvenient and tedious, but if it is not done, using other programs on the same disk may bring about failure of the cargo data programs. No way exists of preventing this danger without modifying the operating system.

6.5.2 File Deletion

No provision is made in the software for deleting the data file for any shipment.

6.5.3 Field Overflow

The storage allocated for the text of such fields as 6-13,56, and 71 is sufficient to contain any input that can be properly written on the input form (one character per box). However, the program does not check actual input to insure that the limits on the use of the field (line length, number of lines, total characters) are not exceeded. Various difficulties can arise if in fact the assumed limits on the field size are exceeded.

6.5.4 Inconsistency In File Access Rights

The data privacy specifications given and implemented in the current version of the program give both shippers and domestic

forwarders access to all data fields for all purposes - enter, change, and read. They are, naturally, limited to reading and editing the data files of only those shipments belonging to them. However, the operator representing a particular shipper can in fact enter any other shipper in the shipper/originator field (Field 6). Unless the shipper as given agrees with the password with which the operator logged into the system, the resulting data file will not subsequently be available to that operator. The same is true for domestic forwarders. The system as presently implemented is subject to accidents. If it were an operational system with such a feature, it would lend itself to abuse - either sabotage or espionage. In legitimate use, it is inconvenient in that it requires input in fields in which only one entry is legitimate. A logical change would be to prohibit (prevent) shippers to enter data into the shipper/originator field explicitly, but rather to have that field filled out automatically when a shipper (identified by his password) creates a file using the ENTER command. The same mechanism should apply to domestic forwarders (agents) and the domestic forwarder field.

6.5.5 Inconsistencies in Field Access Rights

Fields 26 and 28 have provisions for checking "INV" boxes to indicate whether the respective charges should be explicitly included in the invoice. These "INV" boxes are accessible not only for reading but even for writing to the airlines, which would appear to be a violation of the principle that the airlines should have no access to invoice data.

A number of items (see Table 1) are supposedly not available to either the airlines or the government, which are prevented from reading them in the QUERY mode. At the same time, these items are printed on the air waybills, which are available to both the airlines and the government.

6.5.6 Miscellaneous

Field 30 in the input form provides for an INV check mark, to determine whether or not the data should be printed on the commercial invoice. The program currently does not provide for this feature.

Field 47 can not be left blank. Attempts to do so result in error messages until a valid commodity code is entered.

A number of fields to which all legitimate input is either strictly alphabetic or strictly numeric will accept both indiscriminately without producing an error message.

A return to the TOP system (and a cargo data program abort) can result if an operator on the secondary teletype requests data on a shipment to which a file has been assigned (by entering the shipment number - Field 1 - in either the ENTER or EDIT mode), but which has not yet been filed (by terminating entry in either the ENTER or EDIT mode on the primary teletype). This event is unlikely, but it can result in damage to the file directory that is difficult to correct and should not be brought about deliberately.

APPENDIX

SCENARIO FOR DEMONSTRATION ON MARCH 21, 1973

A.1 INTRODUCTION

At the start of the test/demonstration, data terminals will be labeled as follows:

Shipper/Forwarder
Carrier Loading Platform
Carrier Traffic Office
Government Offices

The fifteen cases will already have been entered into the data base of the computer at TSC.

1. Goods on one shipment arrive at carrier loading platform at airport. Airline receiving clerk inputs transaction number, shipper's name, and number of packages. Data prints out in carrier traffic office.
2. Operator at carrier traffic office inputs transaction number and asks for Field 6 to verify the shipper's name.
3. At airline's request, DPC prints out B/L in carrier traffic office.
4. Carrier traffic office assigns shipments to a TWA flight and inputs request for flight manifest - inputs flight number, date, and transaction numbers of each consignment of that flight.
5. Computer advises one shipment has improper destination airport.
6. Carrier traffic office assents to corrected designation.
7. Computer prints out manifest in carrier traffic office and on government terminal.

Note: a. Carrier traffic office instructs computer to transmit to the UK all B/L and invoices related to that specific flight.

b. Sometime during the demonstration acknowledgment of receipt of previously transmitted messages will be received from UK indicating transaction numbers and channel of selection in LACES.

8. Question and answers - queries to data base, etc., etc.

A.2 POSSIBLE QUERIES

1. Airline asks for data on shipment moving on competitive airline (and is rebuffed).
2. Shipper/forwarder requests status information about his consignment.
3. Change unit price and print new invoice.

TABLE A-1. DUN'S CODES AND CORRESPONDING ADDRESSES

00 189 6836	IBM World Trade Corp. 821 UN Plaza New York, N.Y.
21 015 1718	IBM UK Ltd. Havant Plant P.O. Box 6 Langstone Rd. Havant Hampshire, England
21 164 4216	Forwardair RM 5-7/1 Bldg. 521 Cargo Terminal Heathrow Airport London, England
21 015 1627	IBM United Kingdom Ltd. 389 Chiswick High Rd. London W4, England
21 619 1239	BOAC Import Section London Airport London, England
21 015 3656	IBM UK Limited DP-CE Stores 30 Beresford Ave. Wembly, Middlesex, England
21 635 6378	IBM UK Labs Hursley Park Winchester Hampshire England
00 809 2793	Dow Chemical Co. Ltd. Freeport, Texas
21 015 2815	Carryfull Ltd. London Airport North Hounslow Middlesex, England
21 510 0124	Trans Lloyd Ltd. London Airport London, England
21 206 9801	Transparent Paper Ltd. Bridge Hall Mills Bury Lancs, England

TABLE A-1. DUN'S CODES AND CORRESPONDING ADDRESSES (CONT'D)

00 605 2698	Dow Chemical Co. Intl. Inc. (Del.) 1947 Stoutfield West Drive Indianapolis, Indiana
21 008 5510	Dow Chemical Co. Ltd. North Lynn Kings Lynn Norfolk, England
00 653 8268	Dow Chemical Intl. Inc. (Del.) 2020 Abbott Rd. Center Midland, Michigan
00 208 7807	General Electric Co. Intl Sales Division 159 Madison Ave. New York, N.Y.
21 112 1124	Industrial Frt. Long Lane London Airport London, England
21 011 6406	Intl. Gen. Elec. of NY Ltd. Lincoln House 296-302 High Holborn, London WC1, England
21 045 0961	British Steel Corp. for Electrical Stores (Band) Abbey Work PT Talbot Wks PT Talbot Glamorgan S. Wales England
21 045 0979	British Steel Corp. PT Talbot Wks PT Talbot Glamorgan England
21 014 8552	Berwyn Power Equip. Ltd. Bushbury Engr Works Wolverhampton, England
21 013 7527	Hawker Siddeley Dynamics Ltd. Gladstone Rd. Farnworth Bolton Lancashire, England
21 321 4323	Hawker Siddeley Dynamics Ltd. Lostock Bolton England BL6 4BR
21 321 4323	Berwyn Power Equip. Ltd. Howard Shipping Serv. Ltd. Bond House John M. Square Wolverhampton, England

TABLE A-2. IATA AGENT'S CODES AND CORRESPONDING ADDRESSES

0571420	Jet Air Freight 248-06 Rockaway Blvd. Jamaica, N.Y.
3365653	Fort Forwarding, Inc. New York, New York
3385566	J. D. Smith Inter Ocean, Inc. 175-41 148 Rd. Jamaica, New York
2361377	Coughlin Air Romulus, Michigan
1950552	Aerotyme, Inc. New Orleans, La.
4560526	Common Market Forwarders, Inc. Box 60185 AMF IAH Airport Houston, Texas