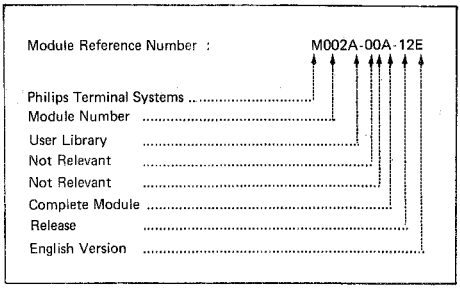
PTS
a programmer's introduction

M2A user library



A publication of
Philips Data Systems
SSS - Training & Documentation
Apeldoorn, The Netherlands

Order Number 5122 993 21132

PREFACE

This manual forms part of the PTS documentation package to support TOSS Release 12. It is designed to provide programmers and systems support staff with an overall introduction to the PTS range of financial terminal systems, the systems in use, the process of developing PTS applications, and the training and documentation provided in support of these activities.

This manual may be used as pre-information, prior to any formal training on PTS, and is used to support course module M100 - Introduction to PTS.

The complete set of PTS documentation to support the user comprises the following modules.

M2A A Programmers Introduction
M4A CREDIT Reference Manual
M5A Device Drivers Reference Manual
M8A TOSS Utilities Reference Manual
M11A DOS-PTS reference Manual
M15A Data Communication Drivers Reference Manual
M21A Programmers Guide - Elementary CREDIT
M22A Programmers Guide - Workstation Handling
M23A Programmers Guide - Disk File Handling
M24A Programmers Guide - Data Communication
M25A Programmers Guide - Work Station Management

For an overview of the complete Training and Documentation package for PTS, please refer to the diagram on the following page.

PUBLICATION HISTORY

This manual was first published in March 1982, based on the initial release of TOSS Release 11.
This version includes an update, published in September 1982, to include information about TOSS Release 12.

PTS Training and Documentation package

Training modules		Training manuals		Reference manuals	
M100	Introduction	M2A	Programmers Introduction		
M110 M111	Elementary CREDIT	M21A	Elementary CREDIT	M4A M91A M5A	CREDIT CREDIT Reference Card Device Drivers
M120 M121	Multitasking in CREDIT	(M21A	Elementary CREDIT)	(M4A (M91A	CREDIT) CREDIT)
M130 M131	DOS-PTS			M11A M90A	DOS-PTS PTS Reference Card
M150 M151 M152 M153	CREDIT Workstation handling	M22A	CREDIT Workstation handling	(M4A (M91A	CREDIT) CREDIT)
M160 M161	Disk file handling	M23A (M21A	Disk file handling Elementary CREDIT)	(M4A M8A (M90A	CREDIT) TOSS Utilities PTS Reference Card)
For basic DC training, refer to the Training Brochure					
M171	Data Communication in CREDIT	M24A	Data Communication	(M4A (M90A M15A	CREDIT) CREDIT) DC Drivers
M190* M191 M192 M193 M194	Workstation Management	M25A	Workstation Management	(M4A (M90A	CREDIT) CREDIT)

Notes : Brackets indicate further use of a module already introduced.
Modules marked with an asterisk are not yet available.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1

INTRODUCTION TO PTS

1.1 The Philips Terminal System

The Philips Terminal Systems are a range of computer systems designed to meet the specialised requirements of banks and other financial institutions. The terminal devices from which the system takes its name allow the bank cashiers (or tellers) to perform their duties with a great degree of flexibility and a minimum of effort, under the control of the Application program in the computer. This degree of flexibility for the operator is achieved without making the programmer's job more difficult.

PTS computers normally operate in individual bank branches, where no specialised knowledge of the system is available. Consequently, the traditional role of the system operator does not exist for these machines. The only operating required is to load the application when the bank staff arrive in the morning (a simple matter of pressing two switches) and to ensure that such things as ribbons and paper in the printers are in a satisfactory condition.

Once an application program has been developed for a bank, it can easily be configured for any branch of that bank, no matter how large or small the branch or how many terminals it has. If required, individual terminal devices may be shared between terminals to give even greater flexibility.

PTS computers in the bank's branches may be connected by data communication lines, either to each other or to a mainframe computer. These connections allow the bank to maintain centralised records of all accounts in each branch.

1.2 Hardware

The complete installation in a branch is controlled by the Terminal Computer, a small but powerful general purpose mini-computer. The terminal devices controlled by this terminal computer are used in two different ways, as Front Office Terminals or as Back Office Terminals.

Front office terminals are those used by the cashiers to perform such tasks as customer enquiries and cash transactions.

Back office terminals are those used to produce information used internally by the bank, such as account listings.

Non-terminal devices such as disk drives and magnetic tape equipment are also available. These devices allow a PTS terminal computer to work either fully independently (maintaining its own files), or to store information for transmission to other computers (for example a mainframe) over data communication lines.

1.3 Software

PTS computers are used in two very different environments. Most commonly, they are found in individual bank branches maintaining records of the accounts in those branches. However, the application programs controlling the sequence of events at the operators' terminals are developed in the more specialised environment of a computer department. As the development software is not required to be present in every branch of the bank, two different operating systems have been developed for use on PTS, one for production and one for development.

1.3.1 Production

The operating system used in the production environment is TOSS, the Terminal Operating System Software. The basis of this operating system is the TOSS Monitor, the controlling program generated independently for each application. The Monitor is responsible for scheduling the tasks that control the sequence of events at each terminal and for maintaining the integrity of the system.

1.3.2 Development

The operating system used in the development environment is DOS-PTS, the Disk Operating System. Under this system, application programs may be written, usually in a dedicated language called CREDIT, translated (compiled) and linked to form an executable program. (Testing of application programs is carried out using the production operating system, TOSS.)

1.4 Setting Up a PTS Application System

Any PTS application program is designed around the functions the cashiers (tellers) have to perform, although other functions are of course necessary for the branch's back office. The choice of equipment in a terminal will be determined by the functions that terminal has to carry out, and will be the same for all cashiers performing the same functions in all branches of the bank.

For example, a cashier handling cash accounts processes only small amounts of information in each transaction. This information is nearly always numeric, so the terminal required for this type of operation need contain only a small display, a keyboard with only numeric and 'control' keys (normally referred to as function keys) and a printer to record details of the transaction.

A cashier handling savings accounts processes similar quantities and types of information, but the printed record of the transaction often involves recording the details in a passbook. Thus, the printer for this type of terminal must be capable of printing on an inserted passbook.

However, in the back office much larger quantities of information are processed, and much of this information is alphanumeric. Larger screens and keyboards are therefore required, together with printers capable of producing extensive reports.

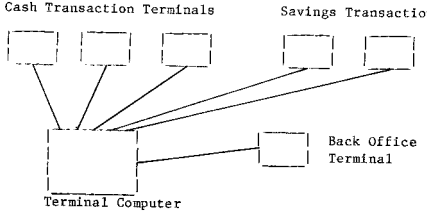


Fig. 1.1 Example Terminal Configuration

Note that the two front office terminal types would probably have the same physical configuration, as the printers available for front office terminals are capable of printing on different types of inserted document. However, as the two types of terminal perform different functions, the system considers them to be different terminal types.

These three different types of terminal would form standard configurations (or Terminal Classes), and this concept is one of the keys to the PTS system. The application software is designed to handle each cashier function via terminals of the appropriate terminal class. The number of terminals which may be included in any particular branch

does not have to be taken into account in any way by the application programmer.

To control and coordinate the activities of the various terminals, a controlling program (the TOSS Monitor) is generated to match the terminal classes defined for the bank's application system. This Monitor will contain only the system software actually required by the application program. Any parts of the system software which are not required (for example for handling devices not included in the application program) are not included in the Monitor, thus reducing its size. Within the Monitor, a `Task Definition' is established for each terminal class.

Finally, the Monitor and application program must be configured for each branch. This involves setting up a `task' in the Monitor for each terminal that exists in the branch. Each task is generated from one of the task definitions included in the Monitor when it was generated. Any number of terminals from any terminal class may be configured, and each will have its own task.

Any terminal device may be configured in more than one terminal. It may even be included in terminals of different terminal classes. Thus, in a lightly loaded branch, for example, a passbook printer may be shared between two cashiers. In a small branch, a general purpose printer may be shared between the front and back offices.

The way the terminals in an individual branch are configured is specified in a data file known as the Configuration Data file. This file contains details of the numbers of terminals in each terminal class and the devices to be included in each terminal.

As consequences of this flexible approach, the programmer need not be concerned about the physical hardware configurations in the various branches, and the end-user (the cashier) can perform any of the defined functions without having to worry about whether the relevant hardware is connected to his terminal.

Chapter 2

PTS HARDWARE

2.1 General Description

PTS hardware can be divided broadly into three categories: the terminal computers themselves, terminal devices and non-terminal devices.

The terminal computers provide maximum memory sizes between 64 kilobytes and 256 kilobytes. A maximum of 32 terminals may be connected to one terminal computer, as well as the range of non-terminal devices.

Terminal devices are the devices used to form the terminals (often referred to as workstations) used by the operators of the system. These devices are normally `dedicated' to one work position in the bank.

Non-terminal devices are devices such as disks and line-printers that are available to be used by any terminal on request, under the control of the application program.

Within this general framework, a further subdivision of terminal devices can be made, as explained in Chapter One. Terminal devices can be intended for use mainly in the front office or mainly in the back office, although there is obviously some overlap between the two areas.

2.2 Front Office Equipment

The devices used to make a Front Office terminal are normally a keyboard, a printer and a display.

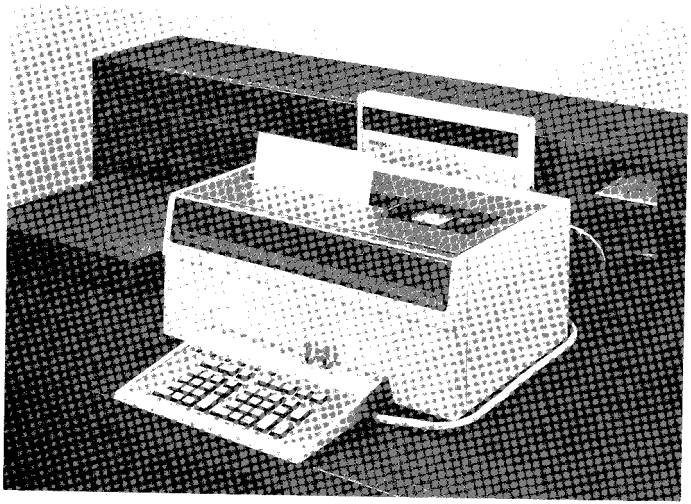


Fig. 2.1 Example Front Office Terminal Configuration

2.2.1 Keyboards

The keyboard may be a full alphanumeric version, incorporating separate numeric and function key clusters, but is more frequently a `numeric' keyboard, so called because it contains only the numeric and function key clusters. The function keys are those used for such things as cursor movement and as end-of-item keys. They do not generate `printable' characters and have no predefined functions. Their meaning is entirely application-dependent.

Both types of keyboard are equipped with lamps which are controlled by the application program. These lamps may be turned on and off or may be flashed, to act as simple prompts for the operator. The keyboards may also have keylocks to allow the application to restrict use of certain types of transaction for security purposes. The application can record

the positions of the locks and prevent unauthorised staff using restricted transactions.

2.2.2 Displays

Like the keyboard, the display used in the Front Office is usually a small version, such as a plasma display. Full-size screens of up to 1920 characters are available, but are normally reserved for use in the Back Office.

2.2.3 Printers

The main output device for a Front Office terminal is the Teller Terminal Printer, a unit containing two or three print stations. One print station (the document station, sometimes referred to as the voucher station) is used to print on pre-printed documents such as pass-books or vouchers. Documents are inserted into the printer by the cashier and are positioned by the application program before printing occurs. Another print station (the journal tape) is for logging the transactions on a standard paper roll for the bank's own internal accounting purposes. In some printers, a third print station (the tally roll printer) is present to print a receipt for the customer.

2.2.4 Badge Card Reader

A Badge Card Reader is available to read the magnetic stripe on a credit card. This may be used to identify the customer without the cashier having to enter an account number from the keyboard. This may allow individual transactions to be handled more quickly.

2.2.5 Financial Terminal

The more commonly-used Front Office devices are also available combined into one unit, the Financial Terminal. This unit contains a numeric keyboard, as described above, a two line (32 characters per line) display, and a journal printer, all in one device. A document printer may be included in the financial terminal if required.

A Personal Identification Number (PIN) keyboard and customer display may be connected to the financial terminal. The PIN keyboard is a small keyboard containing only the ten numeric keys and two function keys, one an enter key, the other a correction key. This keyboard can be used by the customer to enter his own personal (confidential) identification number and other data. The customer display is a one-line display.

The financial terminal may also be equipped with a cash drawer to enable it to be used as a point-of-sale terminal.

2.2.6 Teller Note Dispenser

Another device designed to assist the cashier in his duties is the Teller Note Dispenser. This unit contains up to six different denominations of banknote, new or used or a combination of both. The notes are stored in cassettes within the unit. When a customer withdraws money from his account, the notes are counted and dispensed automatically. This allows the cashier to work faster while ensuring accurate counting of the notes dispensed.

2.2.7 Magnetic Stripe Unit

A magnetic stripe unit is available to read from and write to the magnetic stripes present on credit cards and certain types of passbook.

2.3 Back Office Equipment

Back Office terminals are, like those in the front office, configured using a keyboard, a printer and a display. However, the actual devices present are often different from those used in the Front Office, as the Back Office has different functions to perform.

Fig 2.2 Example Back Office Terminal Configuration

2.3.1 Keyboards

Whereas Front Office terminals may use a numeric keyboard, Back Office terminals will almost certainly use a full alphanumeric version. The transactions processed in the Back Office require more information to be entered, and much of this will be non-numeric information.

2.3.2 Displays

Similarly, the transactions are likely to be more complex in the back office, requiring the operator to deal with larger volumes of information. Therefore, Back Office terminals normally use a full-size Visual Display Unit (up to 24 lines of 80 characters) rather than the smaller displays employed by the cashiers.

2.3.3 Printers

The printing requirements of the two types of terminal are also different. The small sizes of paper handled by the Teller Terminal printers would be inadequate for the often fairly large reports generated in the Back Office. A different type of printer, a General Terminal printer, is normally employed here. This type of printer prints on continuous fan-fold stationery and is thus more appropriate to the needs of this type of transaction.

2.4 Non-Terminal Devices

To complement and support the various terminal devices in both the Front and Back Offices, other devices are obviously necessary. A full range of peripheral equipment is available to allow the computer to operate efficiently.

2.4.1 Disks

To maintain the files of a particular bank branch, disk drives are available for both flexible and hard disks. Flexible disks may be of 256 kilobytes or 1 megabyte capacity, hard disks range from 2.5 to 80 megabytes capacity.

If the PTS computer is sending transaction details to a mainframe computer as they are processed in the branch, it is likely that it does not require its own files. Only an emergency back-up facility is required to deal with problems that may arise if, for any reason, the line is unavailable. In this case, it may be possible to log the transactions on flexible disk, for transmission to the mainframe when the line is restored.

2.4.2 Line Printer

If very large reports are to be produced, a line printer is available. Although a General Terminal printer may be used for such reports, it is a relatively slow device, and the use of a line printer may show a significant increase in speed of processing. However, the line printer is normally reserved for the development environment, where large reports such as program listings are frequently required.

2.4.3 Other Devices

Other devices available primarily for the development environment include magnetic tape drives, a console typewriter and a card reader.

Chapter 3

PTS SOFTWARE

3.1 General Description

As mentioned previously, there are two operating systems on PTS, one each for the production and development environments. The production system, TOSS, allows programs to be tested and run, while the development system, DOS-PTS, allows programs to be written, updated and linked.

3.2 The Production Environment

The production environment is the one most commonly employed for PTS systems. The application program is written at a central computer department, then distributed to all the branches that need to run it. Each branch, therefore, has a completely written and tested program, and no further development need be carried out.

As written at the central site, the application may have to cater for widely differing sizes of branch, from a remote rural office to a large city branch. However, the greatest difference between the requirements of various branches is the number of terminals to be connected. As this value is supplied when the system is loaded (as part of the Configuration Data), the application program itself is identical in every branch. The only difference will be in the configuration data. This makes the system extremely flexible, and if, for example, a branch adds an extra terminal during the life of the application, all that needs to be changed is the configuration data file.

3.2.1 TOSS

The operating system used in the production environment is TOSS (Terminal Operating System Software). TOSS is supplied as part of the system software of the computer. The parts necessary to satisfy the requirements of the particular application program can be selected and linked together to form the TOSS Monitor. As it is unlikely that two different systems will make identical demands on the system software, it can be seen that no two Monitors are likely to be identical.

The TOSS Monitor is generated at the central site in parallel with the application and distributed to the individual branches in the same way. Again, as long as the types of device in the system do not change, the Monitor will not need to be altered during the life of the application. This means it is possible to change the number of terminals in a branch without affecting the Monitor in any way.

3.2.2 Tasks

Both the Monitor and the application program are generated without regard to the number of terminals present in the final system. This information is supplied as part of the configuration data. However, in the running system, each connected terminal must be controlled in some way, for instance to prevent two operators trying to access the same account record at the same time. To enable this kind of control to be exercised, a method of identifying the operations performed by each terminal has been developed.

The actions of each terminal are controlled from within the Monitor. The part of the Monitor responsible for controlling one terminal is known as a task. Thus, there will be one task in the system for each terminal connected. Each task has its own unique identifier.

Other tasks will, however, also be present, as the Monitor uses tasks to perform certain system functions, such as accessing disk files. These tasks are not related to any particular terminal. Their number and types depend upon the way the Monitor was generated.

3.2.3 Terminal Classes

It has already been stated that when the application is written, no account is taken of the number of terminals, and hence the number of tasks, that will exist in the final system. However, those tasks will execute the coding written as the application program, so some way of relating the tasks to the coding of the application program must be available. The Terminal Class is the method of doing this.

All similarly-configured terminals running similar transactions form one terminal class. The application is then written as if only one terminal exists for each terminal class. At run time, when the system is loaded, the configuration data tells the system how many tasks exist for each terminal class. The relevant parts of the coding can then be duplicated to enable each terminal to function correctly.

3.2.4 Data Communication

PTS computers are often connected to other systems by data communication lines. Standard software modules may be included in the Monitor to handle many different data communication protocols from a number of different mainframe manufacturers. These protocols range in scope from a simple point-to-point connection to full networking capabilities.

3.3 The Development Environment

The development environment is the one in which application programs are written. It therefore has different requirements from the production environment and uses a different operating system.

3.3.1 DOS-PTS

The development operating system is DOS-PTS, the Disk Operating System. This operating system allows programs to be written, translated and linked to form an interpretable program. Testing has to be carried out under TOSS.

DOS-PTS is a single-user system that contains all the processors and utilities necessary for application program development and for generating the TOSS Monitor. The various system programs are called into execution using simple commands from the operator's console. The most important commands are those dealing with the language processors.

3.3.2 CREDIT

CREDIT is a language dedicated to the PTS system. As such, it provides a level of flexibility in handling the range of input/output devices in the system which would be difficult to achieve with traditional high-level languages.

The language is interpretive, which means that the object code produced by the CREDIT translator (compiler) is not directly executable, but has to be processed by an interpreter at run-time. This interpreter decodes each instruction to an executable form. This can save considerably on memory size, since the executable code for each instruction need only be held once, in the interpreter, and not every time the instruction is coded.

The application program is written as a series of modules which are subsequently linked together to form the working application. These modules may be available to all tasks in the system, or they may be dedicated to the needs of an individual terminal class.

The translator produces reentrant coding, shared by all the tasks that need to include that particular sequence of instructions. This also reduces memory size, as common coding is only held in memory once.

One module contains a Data Division, where all the data areas for the application are located. The other modules all refer to the data areas

defined in this module. Thus, once the application's data areas have been defined, there is no possibility of invalid data references occurring in the final application.

CREDIT supports four types of data:

Boolean data	one-bit fields, used for holding flags, etc.
Binary data	one-word fields, holding numbers in the range -32768 to +32767 in pure binary format
BCD data	(Binary Coded Decimal data) numeric values of from 2 to 512 digits, each digit held as a four-bit packed decimal value
String data	alphanumeric values of from 1 to 4095 characters, each character held as an 8-bit byte in ISO-7 code

Data items used by a CREDIT program may be either global (available to all tasks) or task local (available to a single task only). Thus, security of data between tasks can be maintained while data needed by all tasks is freely available.

As CREDIT is a language dedicated to PTS, it offers a very wide range of instructions to control the special devices necessary in a banking environment. As well as the instructions for reading and writing information which may be found in any language, CREDIT allows greater control to be exercised over such things as positioning documents in a teller terminal printer, positioning the cursor on a screen and so on.

3.3.3 Screen Management

Since virtually every application written in CREDIT will make extensive use of terminal equipment, and in particular screens and keyboards, a standard package has been written to simplify the programming of such applications. This package is called Screen Management.

Screen Management treats each screen of data as a complete unit and relates all the data fields on the screen to a corresponding data item in memory. The operator can key in the values to each field, either in a predetermined sequence or at random, depending on the requirements of the application. The package performs all the validation necessary. Some checks are incorporated in the package itself, others are included in user-written routines called by the package.

All the statements necessary to accept input from the keyboard and display to the screen are included in the Screen Management routines. This reduces considerably the programming effort required and makes for much easier screen and keyboard handling.

3.3.4 Work Station Management

The Work Station Management (WSM) package is another software package to facilitate workstation handling. As well as the facilities provided by Screen Management, WSM allows the user to have fixed and removable parts on the screen during one transaction, and to display guiding messages for the operator.

The major difference between WSM and Screen Management is that WSM is not included in the user application but runs as a separate task alongside it. The format coding defining the screen layouts is held on a disk file, allowing several user tasks to use the same formats concurrently. Special WSM instructions are included in CREDIT to enable the application to request the functions available.

Formats to be used by WSM are defined with the WSM Creator tool. This is an interactive program which enables the user to design the screen layout while working at the screen, and define the operator input required by specifying a number of parameters for each input field. Error and guiding messages that will be displayed when the related input field is processed may be defined. In addition, a validation routine may be defined for each input field to perform calculations on the input data before it is passed to the application. Validation routines are written in BASIC.

A set of WSM utilities is used for creation and maintenance of the files holding the format code.

3.4 Debugging

To help test and debug CREDIT programs, a special debugging program is available. This debugging program runs as an interactive task alongside the application tasks, and may be used to insert break-points in the coding. When one of these break-points (known as a trap) is reached, execution of the application is halted and the contents of memory locations may be examined and changed, if necessary. This allows small changes to be made to the program during testing without having to go through the lengthy process of updating and retranslating source modules, relinking the application, etc. After the changes have been checked, the necessary updates to the source code must of course be made and the process repeated.

The debugger is not normally included in a tested and running production system.

3.5 Utilities

Two sets of utilities are available for use under the two operating systems.

3.5.1 DOS-PTS Utilities

The DOS-PTS utilities are primarily concerned with keeping the files of each user in a consistent state during the process of developing an application. They allow such precautions as dumps of each user library or of complete disks for backup purposes, and provide the routines necessary to restore a corrupted disk.

A standard utility under DOS-PTS allows four TOSS utilities to be run, to allow a TOSS disk to be formatted, individual files to be created and deleted, and the VTOC to be printed. This facility is necessary when converting an application from DOS-PTS format to TOSS format so that it may be loaded and run.

3.5.2 TOSS Utilities

Whereas the DOS-PTS utilities are mainly concerned with handling groups of files, the TOSS utilities work mainly on individual files. These files are those needed in the production environment and so are mostly data files.

Many of the utilities are used for Data Management files, to maintain these files in a usable condition.

Chapter 4

PTS IN USE

This chapter gives examples of some typical functions which may be performed at a PTS workstation.

4.1 The Front Office

Front office terminals are those which are used to deal directly with the bank's customers, and the devices forming the terminal are chosen accordingly. As was mentioned in Chapter One, the amounts of information processed in the front office are quite small, so normally such a terminal will use a numeric keyboard and a small display.

The most important information produced at such a terminal is the printed record of each transaction. To provide the necessary flexibility in this area a teller terminal printer is normally employed. This allows a record of the transaction to be presented to the customer and also allows the bank to keep a record of each transaction.

The first thing the operator of such a terminal (the bank cashier) would have to do is sign on to the system, usually by entering some kind of identification number from the keyboard. As well as informing the system that the terminal is now working, this procedure allows such operator-dependent information as cash accumulators to be maintained for accounting purposes. It may also restrict the types of transaction the operator is allowed to perform. Similarly, when a cashier has finished at a particular terminal, he should sign off to tell the system the terminal is now vacant.

Once the operator has signed on, he must decide which type of transaction he wishes to execute. The application program will

generally display a screen to inform the operator of the available transactions, and the operator tells the application which of these should be executed by entering a transaction code.

```
ENTER TRANSACTION CODE:

01 CASH A/C DEP.        04 SAV. A/C WITHDR.
02 CASH A/C WITHDR.     05 A/C ENQUIRY
03 SAV. A/C DEP.        06 CURR. EXCHANGE
```

Fig. 4.1 Example Screen Layout for Selecting Transactions

In most PTS applications, the operator identifies the required transaction by pressing a function key, rather than entering a function code, as this facility is available on the keyboards.

4.1.1 Cash Transactions

Many of the transactions carried out in the front office will be concerned with customers depositing and withdrawing cash from their accounts. In general terms, two types of account will exist, cash accounts and savings accounts. The application will contain transactions to handle deposits and withdrawals from both types of account.

If a customer enters the bank and wishes to deposit money on his cash account, the cashier selects the relevant transaction by keying in the transaction code. The application program in the computer would then display the appropriate screen, prompting the operator for the necessary information.

```
CASH ACCOUNT DEPOSIT

ACCOUNT NR.: 99999
NAME: xxxxxxxxxxxxxxxxxxxxxx
AMOUNT: 999,999.99
NEW BALANCE: 999,999.99
```

Fig. 4.2 Cash Account Deposit Screen

In the example in Figure 4.2, the operator would enter the account number and the system would respond by displaying the name of the account holder as a check. The operator would then enter the amount and again the system would respond, this time with the new balance on the account. To terminate the transaction, a further screen could be displayed, as follows.

```
CASH ACCOUNT DEPOSIT

ACCOUNT NR.: 99999

PLACE VOUCHER IN PRINTER
```

Fig. 4.3 Transaction Termination Screen

The operator would place the appropriate voucher in the document station of the teller terminal printer, and the transaction details would be printed, together with a summary on the journal roll for the bank's own internal use.

```
RECORD OF CASH DEPOSITED - CASH ACCOUNT

ACCOUNT NUMBER:  99999      DATE: 99/99/99
NAME: xxxxxxxxxxxxxxxxxxxxxx
BALANCE BROUGHT FORWARD:  999,999.99
AMOUNT DEPOSITED        :  999,999.99
BALANCE CARRIED FORWARD:  999,999.99
```

Fig. 4.4 Print Voucher for Cash Account Deposit

```
TRANS. NR.  9999

TRANS. TYPE 01

A/C NR.     99999

AMOUNT    999,999.99

BALANCE   999,999.99
```

Fig. 4.5 Journal Tape for Cash Account Deposit

After printing, the system may return to the original transaction selection screen.

For savings accounts, the process is basically similar, with the major difference that a passbook is usually used to record the transactions. Instead of printing a voucher, the transaction details would be printed on the next available line of the passbook. The number of the last line printed and the balance remaining when the line was printed can be stored by the application. When the customer wishes to make another withdrawal or deposit, these values can be checked by having the cashier enter the values appearing in the book. So the second screen

for such a transaction could take the following form:

```
SAVINGS ACCOUNT DEPOSIT

ACCOUNT NR.: 99999
ENTER LINE NUMBER: 99
       AND BOOK AMOUNT: 999,999.99
```

Fig. 4.6 Savings Account Deposit Second Screen

If the values shown in the passbook agree with those held by the system, the update may proceed and the final screen may then be displayed.

```
SAVINGS ACCOUNT DEPOSIT

ACCOUNT NR.: 99999

PLACE PASSBOOK IN PRINTER
```

Fig. 4.7 Transaction Termination Screen

The operator would then place the passbook into the document station of the printer. The system would position the passbook to the correct line, the details would be recorded and the transaction terminated, as before.

```
TRANS. NR.  9999
TRANS. TYPE 03
A/C NR.     99999
AMOUNT    999,999.99
BALANCE   999,999.99
```

Fig. 4.8 Journal Tape for Savings Account Deposit

The procedure for cash withdrawals is very similar to that for cash deposits. Only the wording of certain parts of each screen or print line would need changing, as shown in the examples below.

```
CASH ACCOUNT WITHDRAWAL

ACCOUNT NR.: 99999
NAME: xxxxxxxxxxxxxxxxxxxxx
AMOUNT: 999,999.99
NEW BALANCE: 999,999.99
```

Fig. 4.9 Cash Account Withdrawal Screen

```
SAVINGS ACCOUNT WITHDRAWAL

ACCOUNT NR.: 99999

PLACE PASSBOOK IN PRINTER
```

Fig. 4.10 Transaction Termination Screen

4.1.2 Supervisor Functions

For security reasons, certain functions of the front office terminals may have to be restricted to supervisory staff. For example, if the balance or line number on a passbook does not agree with the value held by the application, a correction has to be applied to those internal values. However, the bank may decide that this type of correction is not something an ordinary cashier may perform and that only supervisory staff may make the necessary internal adjustments.

To enable such functions to be performed, PTS application programs may use several methods. One way would be to use the keylock settings on the keyboard. These can be stored by the applicaton, and certain transactions may be restricted to persons holding a particular key. Another way would be for the application to have a number of special operator identities for supervisors. Only these `special´ identities would have access to the restricted transactions.

4.2 The Back Office

The functions performed in the back office are generally more complex and involve larger volumes of information than those in the front office. Consequently, back office terminals will make use of larger displays, alphanumeric keyboards, and printers capable of printing large reports rather than the details of just one transaction.

As with front office terminals, the operator must sign on to the system before any information may be processed.

Once the operator has signed on, the system has once more to be told which transaction is to be run. This could be accomplished in the same way as for the front office, by prompting with a so-called `menu-selection´ screen.

```
                TRANSACTION SELECTION

     01  ACCOUNT ENQUIRY
     02  LIST ALL ACCOUNTS
     03  SET CURRENCY RATES
     04  OPEN SYSTEM
     05  CLOSE TERMINALS
     06  CLOSE SYSTEM

     SELECT TRANSACTION: 99
```

Fig 4.11 Transaction Selection Screen

4.2.1 Transaction Listings

One of the functions of the back office is to produce listings of account details. Two types of listing are often required, details of an individual account or details of all accounts in the system which match certain specified conditions.

Figure 4.12 shows a possible screen layout for details of an individual account. The underlined fields are those which appear at the start of the transaction, the other fields would appear after the account number had been entered.

```
                ACCOUNT ENQUIRY
     ACCOUNT NUMBER 99999

     NAME     xxxxxxxxxxxxxxxxxxxx
     ADDRESS  xxxxxxxxxxxxxxxxxxxx
     CITY     xxxxxxxxxxxxxxxxxxxx

     CURRENT BALANCE 999,999.99

     LAST 3 TRANSACTIONS
     DATE          AMOUNT        DEP./WITHDR.
     99/99/99      999,999.99    D
     99/99/99      999,999.99    W
     99/99/99      999,999.99    D
          MORE TRANSACTIONS? x        PRINT REQUIRED? x
```

Fig. 4.12 Account Enquiry Screen

The question `More Transactions?' could be answered `Y' if the operator wished to see the three preceding transactions on the account, or `N' otherwise, while the question `Print Required?' could be used to obtain

a listing of these account details on the printer (see Fig. 4.13). Alternatively, function keys could be used to control these options, and the prompts could be omitted from the screen layout.

```
|                                                                  |
|                                                                  |
|******************************************************************|
|                                                                  |
|      ACCOUNT NUMBER 99999                                        |
|                                                                  |
|      NAME     xxxxxxxxxxxxxxxxxxxx                               |
|      ADDRESS  xxxxxxxxxxxxxxxxxxxx                               |
|      CITY     xxxxxxxxxxxxxxxxxxxx                               |
|                                                                  |
|      CURRENT BALANCE 999,999.99                                  |
|                                                                  |
|      DATE          AMOUNT        DEP./WITHDR.                    |
|      99/99/99      999,999.99    D                               |
|      99/99/99      999,999.99    W                               |
|      99/99/99      999,999.99    D                               |
|      99/99/99      999,999.99    D                               |
|      99/99/99      999,999.99    W                               |
|      99/99/99      999,999.99    D                               |
|                                                                  |
|******************************************************************|
|                                                                  |
```

Fig. 4.13 Account Enquiry List

If the operator requests a list of all accounts, the system would automatically print the details in the same format as for an individual account list, without displaying anything further on the screen. Such a listing could take a long time, particularly for a large branch with a large number of customers. To allow the terminal operator to continue with other transactions while the listing is being produced, the system could start an extra task. This task would not be related to a particular terminal and would only be active for the length of time necessary to produce the listing.

4.2.2 System Start and Closedown

Before any of the above transactions can be performed, the system must obviously be running. Loading the application is a simple matter of pressing two buttons on the front panel of the computer, but the application may demand that certain other functions are performed before any transactions may be entered.

These other functions may range from the relatively straightforward, such as entering the system date and time, to more complex functions such as setting each operator's cash `float' value and starting each individual terminal.

Figure 4.14 gives an example of a simple screen to perform such actions.

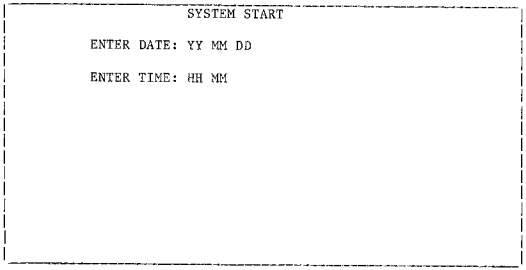


Fig 4.14 System Start Screen

When the day's transactions have been completed, the system has to be closed down correctly to enable processing to resume smoothly the following day. This could be done in two stages, the first to warn the terminal operators that the system is about to be closed and they should therefore not start any new transactions, the second to close the system itself.

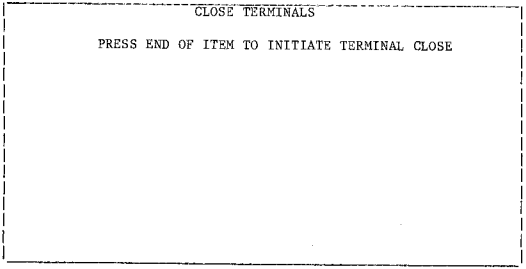


Fig 4.15 Close Terminals Screen

Pressing the End-of-Item function key in response to the prompt in Figure 4.15 would cause a message to be broadcast to each terminal, warning the operators that the system is about to be closed down.

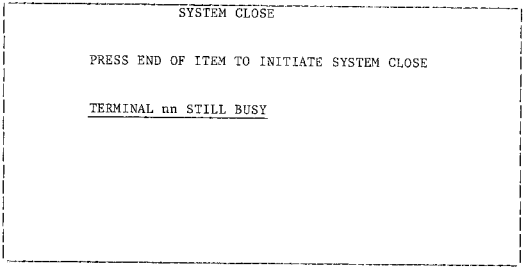


Fig 4.16 Close System Screen

The underlined text in Figure 4.16 could appear if one or more of the terminals had not finished its last transaction. In this case, the System Close function would have to be repeated later to close the system correctly.

4.3 Communication with Computer Centre

PTS computers are often used to pass information to another computer by means of a data communication link. Software modules are available to support a number of standard data communication protocols, and may be included in the Monitor as required.

Data may be transferred in batches, or it may be transferred as the transactions are processed. The latter is known as real-time transmission. Batch transfer means that all the information is collected by the PTS machine and stored on disk, then transferred to the mainframe after normal processing has been completed for the day.

4.4 System Integrity

To maintain the validity of the data stored and processed by a PTS computer, several features are built into the system.

In the event of failure of the processor power supply, the TOSS Monitor is able to store certain vital information in the short interval between detection of the failure and total power loss. When power is restored, other routines in the Monitor re-try the last operation on every active I/O device, including the terminal devices. Thus, the system can continue from the point at which the failure occurred with no loss of data.

If a power failure occurs at any of the terminal devices, optional routines may be included in the Monitor to complete any outstanding request to the device with an error condition. This allows the

application program to perform its own error and recovery routines. If the completion routines are not included in the Monitor, any outstanding I/O requests will be completed as normal, with no error indication.

To preserve the integrity of data files, magnetic tape drives may be included in the system and dumps may be made of critical data at regular intervals, either under the control of the application program or by a free-standing utility program.

If the PTS computer is sending data in real-time to another computer, it is possible for the communication line to become inoperable while the system is running. In this case, the application program in the PTS computer can switch to `off-line' mode. The details of any transactions occurring while the line is unavailable can be stored on flexible disk. When the line becomes operable again, these details may be forwarded to the mainframe. Depending on the functions involved in the application program, this may involve some loss of functions, but does allow many transactions to be performed without the link to the mainframe (or host) machine.

For example, if the mainframe holds the main data file for the application, enquiry transactions will not be able to be processed while the line is unavailable. Cash deposits may continue to be handled, and the details sent on to the mainframe later.

Chapter 5

PTS APPLICATION DEVELOPMENT

The development of a complete PTS application such as was outlined in Chapter 4 involves the creation of three distinct parts:

- the application program itself
- the TOSS Monitor
- the Configuration Data file

All three parts are created under the development operating system, DOS-PTS, then transferred to a medium suitable for loading under TOSS. The resulting application software is tested under TOSS. Any errors detected may be corrected by repeating the process until the application works correctly.

Development is usually carried out on a large hardware configuration, as the system software exists on hard disk and requires such things as a line printer to be available. Testing may be carried out on the same machine, or another machine may be available for this purpose, supporting only the terminal devices necessary for the application.

The development process is summarised in figure B.1 in Appendix B. This may be folded out for use with the more detailed explanation given in the rest of this chapter.

5.1 CREDIT Application Development

The CREDIT application program is written as a series of modules, one of which contains the global Data Division. All the other modules in the application refer to this Data Division for their data definitions. Every module contains a Procedure Division, where the CREDIT instructions to perform the various functions required by the

application are coded.

Once they have been coded, these modules have to be input to the system. Input is normally via the console typewriter, although punched card or magnetic tape input can also be used. Each module is then translated to obtain the object code to be processed at run-time, and finally all the object modules are linked together to form the complete application program.

5.1.1 CREDIT Translator

Each module must be individually translated to check for syntax errors and to generate the intermediate object code. This is the code which will ultimately be interpreted at run time.

The translator produces three listings. The first is the source listing, containing all the source code as entered to the system, and details of any syntax errors it may contain. The other listings are of all the data items defined in the Data Division and all labels declared in the Procedure Division.

If no syntax errors are present, the intermediate object code is produced and may be kept for use in the next stage of development.

5.1.2 Updating CREDIT Source Modules

If any syntax errors occur during translation, or if any logic errors are discovered, the source modules must be updated. This can be done by using one of two processors, the Line Editor or the Text Editor.

The Line Editor is a sequential processor for updating source or data files. Each line of the file has a unique sequence number associated with it, and the Line Editor processes these lines in the sequence of their numbers.

The Text Editor is a more complex processor which offers greater flexibility in updating the file. The lines may still be identified by their sequence number, if required, but they may also be referred to by any character string they contain. This allows updates to the file to be made in any sequence.

Once the source modules have been updated, they may be kept and retranslated.

5.1.3 CREDIT Linker

The CREDIT Linker takes the Intermediate Object Modules produced by the CREDIT Translator and links them together to form one final object module. At the same time, it solves some of the references the Translator cannot solve for itself.

The Intermediate Object Modules produced by the Translator may contain two types of unresolved references: those to other CREDIT routines and those to system routines. The CREDIT Linker solves the first type of

reference, references to system routines are solved by the Linkage Editor.

5.1.4 Linkage Editor

The output object file from the CREDIT Linker is then input to the Linkage Editor. The Linkage Editor will link in any system routines the application requires.

The major system routine included by the Linkage Editor is the CREDIT Interpreter. This is the part of the application program which `decodes' the object code at run-time and executes each instruction. It must therefore be present in every CREDIT application.

Also included at this stage, unless it is specifically excluded, will be the CREDIT debugger to enable the application to be tested.

All these modules will be linked together to form one program, the application Load Module.

5.2 Generating the TOSS Monitor

The second major component of the complete system is the TOSS Monitor. The complete Monitor is supplied as part of the system software. If it were to be used complete in every application, it would be very wasteful of memory, as many of the functions would never be needed. Therefore, the relevant parts of the Monitor for the application in question are selected and linked together to form the Monitor required. So if the application does not make use of a general terminal printer, for example, the software for this type of printer will not be included and the size of the Monitor will be reduced.

The program to perform this selection and linking procedure is a standard DOS-PTS system program, SYSGEN (SYStem GENeration). This is an interactive program that is run from the console typewriter. It uses the responses to a series of questions to generate the parameters necessary to select the relevant Monitor modules.

Output from the SYSGEN program is a Monitor Load Module.

5.3 Generating the Configuration Data File

The final element needed to create a running system under TOSS is the Configuration Data file. This contains details of the numbers of terminals connected for each terminal class within the application, and other information necessary to run the system successfully, such as the priority (the relative `importance') of each task.

The Configuration Data file is created as an ordinary data file under DOS-PTS. If necessary, it can be updated by using the Line or Text Editor, in much the same way as a source file.

5.4 Transferring from DOS-PTS to TOSS

All the procedures outlined so far in this chapter take place under the DOS-PTS operating system, but the resulting files have to be used under TOSS. As the two operating systems use different disk formats, some method must be available to effect the transfer from one operating system to the other.

Loading a TOSS system is performed from disk, either hard disk or flexible disk, and a DOS-PTS procedure ($PDISC) is available to make the transfer.

All that is then necessary is to load the device into the appropriate drive and follow the IPL (Initial Program Load) procedure for the machine. The application program will be loaded. If the CREDIT Debugger has been included, this will gain control, otherwise the application itself will begin.

5.5 Testing

When the complete application has been transferred to a TOSS-formatted disk, it may be loaded into a machine and tested. To make testing easier, the CREDIT debugger is available. The debugger is included automatically by the CREDIT Linker, unless it is specifically excluded (e.g. for a production system).

If any errors are found during testing, the source modules must be updated and the application relinked before a further test is carried out.

Chapter 6

PTS TRAINING AND DOCUMENTATION

To obtain detailed knowledge of all PTS subjects, a Training and Documentation package has been developed. This package ranges in scope from the basic information necessary for people with no knowledge of the system to the advanced and very detailed information required by the software specialist. To cover this very wide range of information, the package has been developed as a number of modules.

Figure B.2 in Appendix B shows a list of all training and documentation modules currently available for the CREDIT programmer. It is reproduced in the Preface of every PTS manual.

6.1 Module Types

The package contains three types of module:

- Reference Manuals
- Training Manuals
- Training Modules

The Reference Manuals contain detailed information about such things as syntax rules for language statements or parameters for running a utility. They are intended to provide the complete facts about PTS for the experienced programmer and to support the Training Manuals and Training Modules. They are not intended to be self-explanatory for a newcomer to PTS, so the Training Manuals and courses provide guidance in their use.

The Training Manuals take the form of Programmers Guides. They are intended to show the programmer how to use the facilities defined in

the Reference Manuals to achieve the results he wants. They provide descriptions of the way the various software components work and provide cross-references to the relevant Reference Manuals. They are intended to be used by inexperienced programmers, or by experienced programmers who need to know more information about parts of the system they have not used before.

The Training Modules form a basis for a series of courses on PTS equipment, its programming and use. These courses make extensive use of the Reference Manuals and Training Manuals (the Programmers Guides) and thus provide experience with their use.

The Training Modules shown in Figure B.2 are not necessarily usable independently. Each course normally makes use of a combination of these modules.

6.2 CREDIT Application Development

When writing a CREDIT application program, two types of information are required: information about CREDIT itself, and about the DOS-PTS operating system.

6.2.1 CREDIT

The CREDIT Reference Manual, Module M4A, provides detailed information about each CREDIT statement, its syntax and an example of the way the statement should be coded.

The Reference Manual does not provide examples of sequences of instructions to solve particular problems, or general information about the PTS system. This type of information is available in the PTS Programmers Guide, a series of modules providing information on different aspects of programming for PTS applications:

- Module M21A is called Elementary CREDIT. It supplies information about the statements likely to be found in any CREDIT application, as well as more detailed information about the programming of each of the devices that may be connected to the system.

- Module M22A is called Workstation Handling. It gives details of two more-advanced methods of handling screens and keyboards in a CREDIT application, Format Input/Output Control and Screen Management.

- Module M23A is called Disk File Handling. This module gives details of the three Data Management packages available in CREDIT.

Finally, a quick reference summary of the CREDIT language is provided as Module M91A, the CREDIT Reference Card.

A course in CREDIT programming may make use of several training modules. Assuming the students have no prior knowledge of either CREDIT or PTS, such a course may begin with module M100, an Introduction to PTS. This may be followed by modules M110, Introduction to CREDIT and M111, Elementary CREDIT. Modules M112, [illegible] M120, Multitasking in CREDIT and [illegible]

dealing with such subjects as disk file handling and workstation handling may be included at this stage or may be left until later, when the students have more experience with CREDIT.

6.2.2 DOS-PTS

Information about the DOS-PTS operating system is contained in Module M11A, the DOS-PTS Reference Manual. This manual contains the detailed syntax of all the DOS-PTS commands and descriptions of the processors used in creating an application for a PTS computer, such as the CREDIT Translator, the CREDIT Linker and the Linkage Editor.

For quick-reference information about the DOS-PTS and TOSS operating systems, Module M90A, the Programmers Reference Card, is available.

Training modules M130, Introduction to DOS-PTS and M131, DOS-PTS, give details of the DOS-PTS operating system. This may be given as a 'free-standing' course, or may be given as part of a CREDIT course.

6.3 Generating the TOSS Monitor

Details of the program SYSGEN, used to generate the TOSS Monitor, are contained in Module M12A, Monitor Generating and Configuration. This manual also contains details of the Configuration Data necessary to configure the tasks within the TOSS Monitor at run time.

A number of software routines to control the hardware devices in the system are contained within the Monitor. These routines are known as the Device Drivers, and details of these Drivers are available in Module M5A, the Device Drivers Reference Manual.

Training modules M140 (Introduction to Monitor Generation) and M141 (Monitor Generation and Configuration) give details of how to run the SYSGEN program and how to configure the system. They may be given as part of a more general course.

6.4 Data Communication

Training and documentation modules provided for data communication subjects range from the basic level of introductory courses and manuals to the specific software requirements and capabilities of the PTS system itself.

Module A13, Introduction to Data Communication, provides an introduction to data communication principles and terminology.

Details of the CREDIT statements available for data communication are in Module M4A, the CREDIT Reference Manual. More detailed explanations of the use of these statements will be found in the Programmers Guide for Data Communication, Module M24A.

Details of the Drivers available for the various data communication protocols can be found in Module M15A, the Data Communication Drivers Reference Manual.

A standard software package is available to allow a PTS computer to simulate an IBM3270 terminal. Details of the generation and use of this package may be found in Module M19A, the IBM3270 Simulator Reference Manual.

Training modules available for data communication include modules A100 to A102 (Introduction to Data Communication) and A110 (Introduction to Networks). On PTS subjects, module M171 covers Data Communication in CREDIT and M172 the IBM3270 Simulation Package.

6.5 Testing

The main tool used for testing a CREDIT application is the CREDIT Debugger. Details of the Debugger commands may be found in the CREDIT Reference Manual.

For creating and maintaining disk files under TOSS, a series of utilities are available. Details of these utilities are in Module M8A, the TOSS Utilities Reference Manual.

6.6 Use of Documentation Modules in the Development Process

When writing a CREDIT application, information is required at several levels. An experienced CREDIT programmer would probably only need the CREDIT Reference Card to check syntax rules for individual statements, with occasional reference to the Reference Manual for more complex or infrequently-used statements. A less experienced programmer, or an experienced programmer working in a new situation, may need to refer to the Programmers Guide for more detailed explanations of the statements, or to find out which statements to use to solve a particular problem.

Once the application has been written and is being tested, similar information is again often required. In this situation, the Reference Cards can usually supply the necessary facts (e.g. the correct syntax for a particular statement in the case of a syntax error), although reference to the relevant manual may be necessary in some cases.

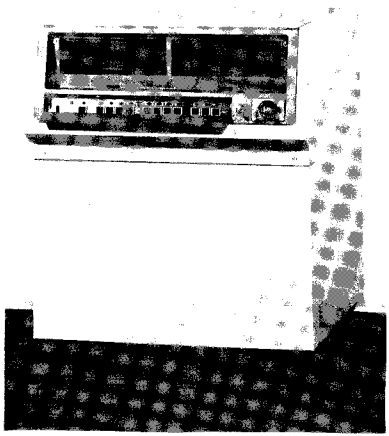
APPENDIX A

PTS DEVICES

This Appendix contains descriptions of some of the more commonly-used PTS computers and devices.

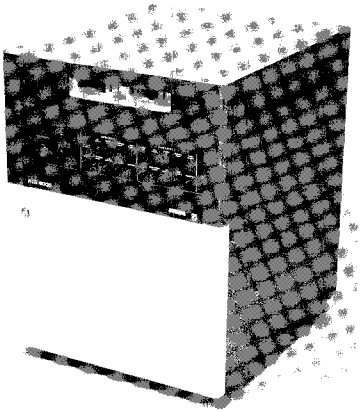
A.1 Terminal Computers

PTS6805



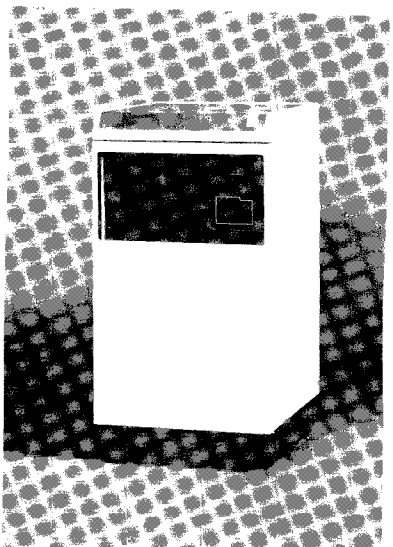
- 64 kbytes memory
- flexible disk drive
- up to 6 terminals

PTS6810



- max. 64 kbytes memory
- cassette drive standard, flexible disk available
- up to 16 terminals

PTS6813



- max. 256 kbytes memory
- flexible disk standard, cassette drive available
- up to 32 terminals

A.2 Terminal Devices

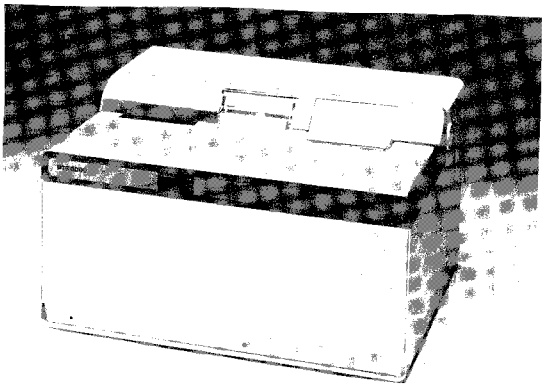
Although there is no fixed division between front and back office terminal devices, these headings are used to match the examples given in the text.

A.2.1 Front Office Terminals

PTS6271 Numeric Keyboard

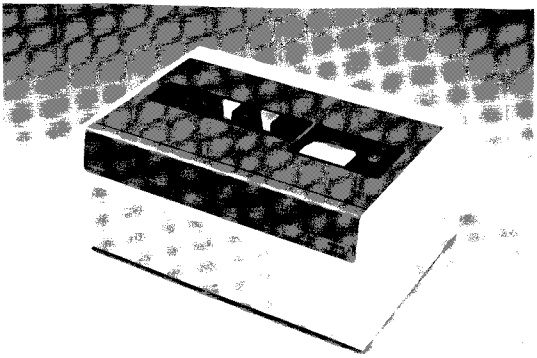
- 6 lamps (program controlled)
- 2-4 keyswitches
- 12 numeric keys (including double- and multiple-zero keys)
- 28 function keys

PTS6225 Teller Terminal Printer



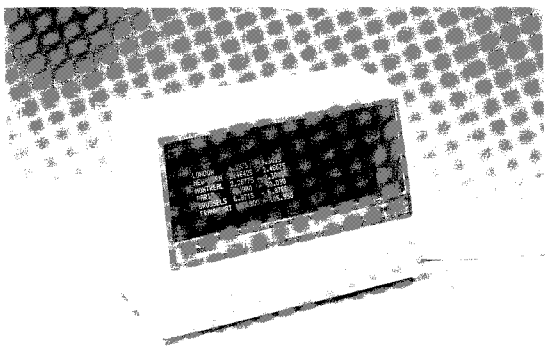
- journal printer
- document printer
- tally roll printer
- 9 national character sets (hardware controlled)

PTS6371 Teller Terminal Printer



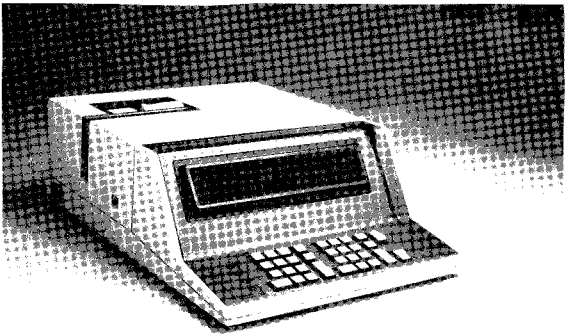
- journal printer
- document printer
- character set program controlled
- document parameters program controlled
- document station handles documents up to A4 size

PTS6386 PDU



- 6 lines of 40 characters
- 10 character sets, hardware controlled

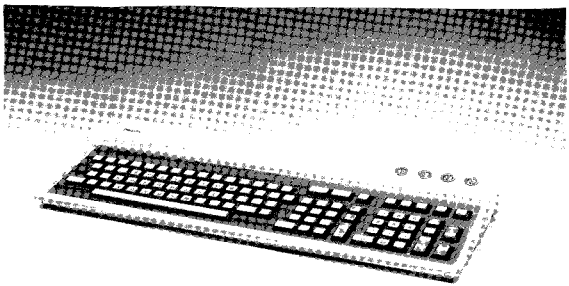
PTS6280 Financial Terminal



- 32-key numeric keyboard
- 4 lamps
- display of 2 lines by 40 characters
- badge card reader
- journal and document printers

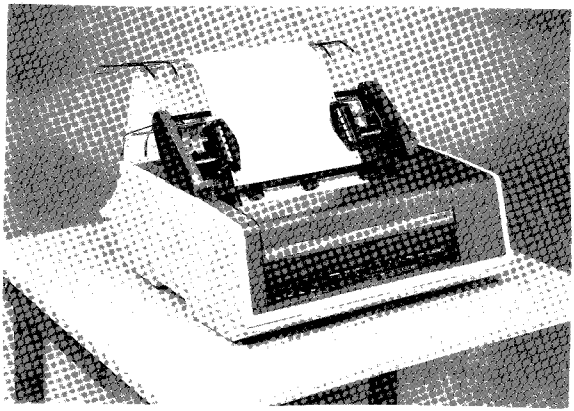
A.2.2 Back Office Terminals

PTS6272 Alphanumeric Keyboard



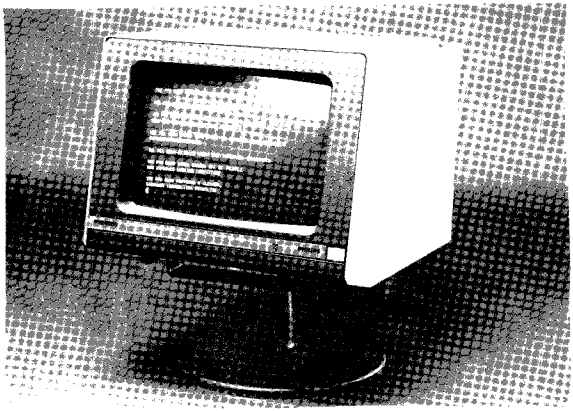
- 6 lamps (program controlled)
- 2-4 keyswitches
- 12 numeric keys (including double- and multiple-zero keys)
- 59 alphanumeric keys (inluding function, shift and control keys)
- function key cluster of 28 keys

PTS6374 General Terminal Printer



- character set, character font, line spacing, etc. all software controlled
- 2 modes: normal quality, high speed or
  high quality, low speed
- tractor feed and front feed options

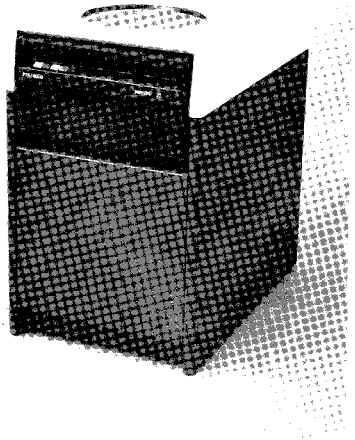
PTS6344/6346 VDU



- 20 lines of 64 characters or 24 lines of 80 characters (hardware adjusted)
- upper and lower case
- underline
- high/low intensity

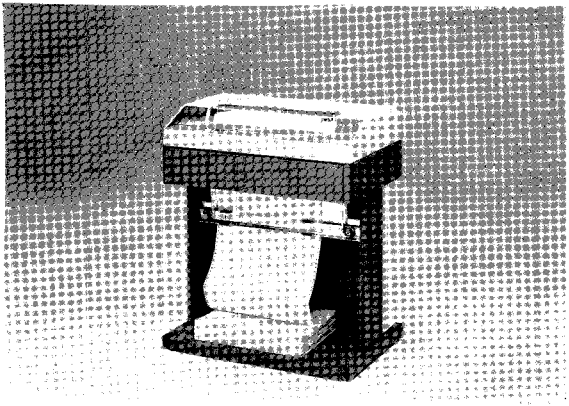
A.3 Non-Terminal Devices

PTS6876 Disk



- fixed and removable cartridge disks in one drive
- both disks 5 mbyte capacity

PTS6882 Line Printer



- 400 lpm printer
- 132 characters/line

APPENDIX B

TABLES

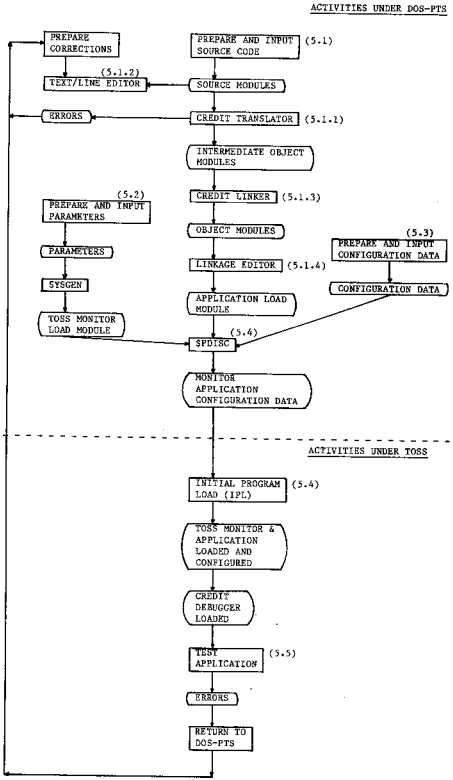


Fig. B.1 CREDIT Application Development

PTS Training and Documentation package

Training modules		Training manuals		Reference manuals	
M100	Introduction	M2A	Programmers Introduction		
M110 M111	Elementary CREDIT	M21A	Elementary CREDIT	M4A M91A M5A	CREDIT CREDIT Reference Card Device Drivers
M120 M121	Multitasking in CREDIT	(M21A	Elementary CREDIT)	(M4A (M91A	CREDIT) CREDIT)
M130 M131	DOS-PTS			M11A M90A	DOS-PTS PTS Reference Card
M150 M151 M152 M153	CREDIT Workstation handling	M22A	CREDIT Workstation handling	(M4A (M91A	CREDIT) CREDIT)
M160 M161	Disk file handling	M23A (M21A	Disk file handling Elementary CREDIT)	(M4A M8A (M90A	CREDIT) TOSS Utilities PTS Reference Card)
For basic DC training, refer to the Training Brochure					
M171	Data Communication in CREDIT	M24A	Data Communication	(M4A (M90A M15A	CREDIT) CREDIT) DC Drivers
M190 M191 M192 M193 M194	Workstation* Management	M25A	Workstation Management	(M4A (M90A	CREDIT) CREDIT)

Notes : Brackets indicate further use of a module already introduced.
Modules marked with an asterisk are not yet available.

Fig. B.2 PTS Training and Documentation Package