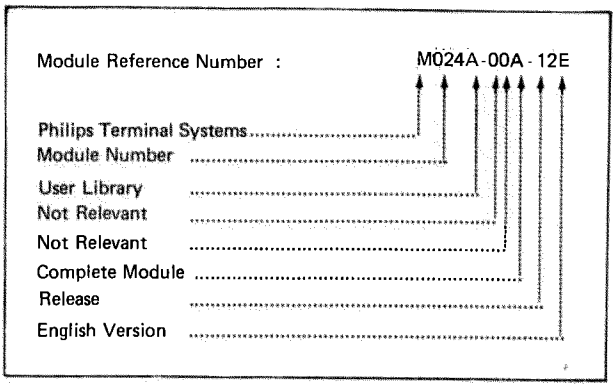
PTS

programmer's guide data communication

M24A user library

Data Systems

PHILIPS



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

Order Number 5122 993 20532

PREFACE

This manual forms part of the PTS documentation package to support TOSS Release 12. It describes all the CREDIT statements to enable the user to include Data Communication handling in a CREDIT application, and gives an example of each one. Its use is defined at two levels:

It is designed as reference material for the less experienced programmer, and is also used to support course module M171.

Readers of this manual are assumed to be familiar with the principles of elementary CREDIT, and with those of Data Communication.

For guidance on the other parts of CREDIT programming, involving elementary CREDIT, workstation handling and disk file handling, refer to the other parts of the Programmers Guide, modules M21A - M25A.

For guidance on how a program for PTS is developed, refer to module M11A, DOS-PTS Reference Manual.

For guidance on the basic principles of Data Communication, refer to module A13, Introduction to Data Communication.

The complete set of PTS documentation to support the user comprises the following modules. Modules related to the subject matter in this manual are marked with an asterisk.

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
M90A	PTS Reference Booklet
*M91A	CREDIT Reference Booklet

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

PUBLICATION HISTORY

This version, published in October 1982, is a complete new version based on the initial release of TOSS Release 12.

CONTENTS

Chapter 1

INTRODUCTION

The CREDIT Programmers Guide is designed to complement the training courses, and can also be used as a self study guide for application programmers to acquaint themselves with CREDIT on the Philips Terminal System (PTS). It gives the reader information for further reading in the form of cross-references to other PTS documentation.

This module introduces the subject of CREDIT programming for Data Communication applications. Chapter 2 describes the statements necessary for handling Point-to-Point connections, and gives examples of each one. Chapter 3 describes the statements for Multipoint connections, and gives an example of each one. Chapter 4 extends this information, with regard to connections to networks.

Note: the various options described in Chapters 2 and 3 may not apply to all Data Communication Drivers, but are based on drivers DRDC17: BSC Point-to-Point, and DRDC07: HDLC Multipoint. If any other driver is used, reference should be made to module M15A, Data Communication Drivers Reference, for details of the options available for each instruction.

In addition, while it is possible to write the necessary code to handle DC in a CREDIT application, by use of this module together with M4A, the reader is advised to refer to M15A if more details on the driver actions are required.

In the chapter on Point-to-Point the term DLS (Data Link Station) is used. This term refers to the software/hardware interface composed of the driver and the channel unit. Each physical DC line constitutes one DLS, so one computer may contain more than one DLS.

When discussing multipoint and network connections, the term DTE (Data Terminal Equipment) is generally used for the end-point of a data link, and this term is used in the Chapters 3 and 4, instead of DLS.

It is only important to the CREDIT programmer to know what mode or state the DLS or DTE can be put into (e.g. "Active", "Disconnected") as a result of each instruction, and this is described in the following chapters.

At the foot of appropriate pages in this guide the CREDIT instruction mnemonics are shown, to provide a quick reference to module M4A, CREDIT Reference Manual.

Chapter 2

POINT-TO-POINT

2.1 Introduction

A point-to-point connection normally consists of the following:

The PTS system is connected, via a modem, to a telephone line (leased or switched), and, via another modem, to another computer at the other end of the line. As far as the CREDIT programmer is concerned, it is not important what type of computer is connected at the other end; it may be another PTS system, or a large mainframe.

In addition, the Monitor software (driver) is designed to handle the protocol that is being used to communicate with the other system. This means that the CREDIT application only has to set up messages that it wishes to transmit, for example, and the driver will ensure that the correct control characters are added, and that the message is acknowledged by the other system, and so forth.

However, the CREDIT application must be able to use the line, and must therefore `open' the communication before starting any transmission. It may also `close' the communication when it does not need to use it for some time.

The general sequence of events that must be controlled by the application are as follows:

Opening the physical connection
Establishing the logical connection
Transmitting messages and/or
Receiving messages
Closing the logical connection
Closing the physical connection if no longer required

The application may set a timeout value for the operations reqired to make logical connection and to receive messages, in the same way as attaching a device. This is done by means of the DSC instruction with Control code X'0B', as described in module M4A, CREDIT Reference Manual. Note that the timeout value set must allow for transmission retries performed by the driver.

In addition, the DSC instruction can be used for handling status and statistical information, as described in section 2.8.

In the following sections, the CREDIT statements for carrying out the communication over the line are described, together with those for timeout and statistic handling.

2.2 Opening the Communication Link

The OPEN .DC instruction is used to perform the physical connection of the computer to the communication line, since after IPL the DLS is considered to be inactive until this request is succesfully completed.

Note that, in the case of a switched line, the application must first issue a DSC instruction to Set Status, as described in section 2.8.1.

The instruction may be issued with Wait or No Wait, as required by the application.

No parameters are necessary for the connection to be made, but one binary data item (dummy) must be coded to ensure the correct syntax.

The Condition Register will be set to one of the following values as a result of this instruction:

CR Value	Meaning
0	OPEN successful
2	Error condition

If the request is successful, the DLS is considered to be in Disconnected mode (i.e. Active and physically connected, but not logically ready for transmission). Before any messages can be sent or received, the line must be logically connected, as described in the following section.

The only error that can occur is Physical Connection Error, which may be caused by one of:

- The modem being not operable
- The channel unit not being connected to the computer
- An incorrect channel address

The reason can be established by fetching the status word from the driver, by issuing an XSTAT instruction. For the meanings of the bits in the status word, reference should be made to the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

In any of the above cases, the DC driver will keep trying to establish the physical connection until a CLOSE is performed. It is recommended that the application issue a CLOSE request and repeat the OPEN request until the result is successful, indicated by the Condition Register being set to zero.

| OPEN .DC |

Example of the OPEN instruction

```
OPEN  .DC,DSLIN,DUM
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DUM This is a binary data item, included for syntax reasons. Its contents are not significant.

```
-------------
| OPEN .DC  |
-------------
```

2.3 Logically Connecting the Line

The CONNECT statement is used to perform the logical connection of the DC line, as follows:

After physically opening the communication line, it is the responsibility of the application to perform a logical connection, to indicate to the system at the other end of the line whether it wishes to send or receive data. After transmission in one direction (as many times as necessary), if it is required to reverse the direction of the messages, the line must be logically disconnected, and a new logical connection must be made with the correct parameters to ensure that the messages are now going in the opposite direction.

Thus, for example, an application wishing to send an account number or similar to a mainframe, and then receive the balance of the account, must first connect in Active mode, meaning that it is the `master' of the exchange and wishes to send a message. Following successful transmission of the account number, it must then disconnect and reconnect in Passive mode, meaning that it is now the `slave', and wishes to receive a message, in this case the balance of the account.

The CONNECT .ACT (active) or CONNECT .PAS (passive) instruction is used, depending on the application requirements, but in either case the syntax of the statement operands is the same.

The instruction may be used with Wait or No Wait, as required by the application.

No parameter information is required, but a dummy binary data item must be coded for syntax reasons.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	OPEN successful
2	Error condition
3	ENQ received (only for CONNECT .ACT)

If successful, the DLS is considered to be in the Connected mode, i.e. it is able to send or receive messages, depending on whether it performed an Active or Passive connection.

CONNECT .ACT
CONNECT .PAS

If ENQ is received, it indicates that the system at the other end of the line wishes to send. The application should disconnect and connect again in passive mode as soon as possible (assuming that the system design has determined that the other system has priority).

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are driver dependent, and are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

Example of the CONNECT instruction

```
CONNECT .ACT,DSLIN,DUM
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class. In this example, the keyword .ACT indicates that the application wishes to send a message, as described above.

DUM This is a binary data item, included for syntax reasons. Its contents are not significant.

```
| CONNECT .ACT |
| CONNECT .PAS |
```

2.4 Sending a Message

To send a message on the line, after performing the correct sequence of OPEN and CONNECT .ACT instructions, the SEND instruction is used.

The SEND instruction requires three parameters to specify the data item containing the message, the length of the message, and the mode and frame ending character options.

The instruction may be issued with or without Wait, as required by the application.

More than one SEND instruction can be issued by the application, as required, but if it is then desired to receive a message, the line must be logically disconnected with the DISCNCT instruction, and reconnected for message reception with the CONNECT. PAS instruction. Thus the normal messages are sent with the option set to 0 or 2, to be terminated with ETB. The last message before disconnecting should be sent with ETX (option 1 or 3), and this should then be followed by the disconnect request (see 2.6).

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	SEND successful
1	Too many WACKs
2	Error condition
3	RVI received

If too many WACKs have been received in reply to the request, due to the mainframe being busy with some other processing, the DLS switches to Disconnected mode, and a logical Connect must be reissued, to restablish connection. Note that the maximum number of WACKs is set in the system software, and can not be influenced by the application.

If RVI is received, this indicates that the other system wishes to send a message. The application may continue to send messages if required, using the ETX option for the last one. It must then disconnect and reconnect in passive mode, and then issue a receive request to read the message from the line.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| SEND |

Example of the SEND instruction

```
SEND  .NW,DSLIN,MESBUF,MESLEN,OPTNS
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

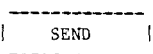
MESBUF This is a string data item, containing the message to be sent on the line. The driver will add the necessary framing characters before transmission. However, the mode and the frame ending character must be specified by the application (see OPTNS below).

MESLEN This is a binary data item, containing the length of the message to be sent. Thus the string data item may be longer than the actual message, as the number of characters sent is determined by the value in this data item.

OPTNS This is a binary data item containing the option for Transparency and Frame ending character, as follows:

Value 0 = Non-transparent mode, Frame ending character ETB
Value 1 = Non-transparent mode, Frame ending character ETX
Value 2 = Transparent mode, Frame ending character ETB
Value 3 = Transparent mode, Frame ending character ETX

Note: Transparency is only possible if EBCDIC and transparency was specified during Monitor generation. If Transparency is specified and EBCDIC is not used, the request will be completed with the Condition Register indicating Error (value 2).



2.5 Receiving a Message

To receive a message on the line, after performing the correct sequence of OPEN and CONNECT .PAS instructions, the RECEIVE instruction is used.

The RECEIVE instruction requires three parameters, specifying the data item to contain the message, the length of the message, and the reply to be sent to the last received message. This last option is contained in a binary data item, and may be set to 0 if the reply is to be ACK0/1 (normal transmission), or 1 if the reply is to be RVI.

The instruction may be issued with or without Wait, as required by the application.

The RECEIVE may be repeated several times by the application, as required, but if it is then desired to send a message, the line must be logically disconnected with the DISCNCT instruction, and reconnected for message transmission with the CONNECT. ACT instruction.

In receive mode, the line cannot be disconnected when the application wishes to send a message, as the other system is master, and this might result in loss of data. Therefore a RECEIVE must be issued with the option set to 1 to indicate that the application wishes to send. A further RECEIVE, or possibly more than one, must be issued, until one is completed with the Condition Register set to 1, when the EOT has been received. Then the application may safely disconnect, and reconnect in Active mode, in order to send one or more messages.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	RECEIVE successful
1	EOT received
2	Error condition

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

RECEIVE

Example of the RECEIVE instruction

```
RECEIVE .NW,DSLIN,MESBUF,MESLEN,OPTNS
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

MESBUF This is a string data item, into which the message will be read from the line. The driver will remove any framing characters before placing the message in this data item.

MESLEN This is a binary data item, containing the maximum expected length of the message to be read. Thus the string data item may be longer than the actual message, as the number of characters read is determined by the value in this data item. If the message is longer than this value, an error condition will be reported via the Condition Register.

OPTNS This is a binary data item containing the option indicating that ACK or RVI should be sent in reply to this message, as follows:

Value 0 = Reply with ACK0/1 to the received message.
Value 1 = Reply with RVI to the received message.

2.6 Logically Disconnecting the Line

The DISCNCT statement is used to perform the logical disconnection of the DC line, as follows:

After opening the communication line, and performing the logical connection, as described previously, messages may be sent or received. After transmission in one direction (as many times as necessary), if it is required to reverse the direction of the messages, the line must be logically disconnected, and reconnected again with the correct parameters to ensure that the messages are now going in the opposite direction.

Thus, in the example of the application wishing to send an account number to a mainframe, and then receive the balance of the account, the logical disconnection and reconnection must take place between the sending of the account number, and the reading of the balance message.

The DISCNCT instruction is used to perform the logical disconnection, either for the purpose of changing the direction of the message transfer, or if the application no longer requires the line, in which case it must be followed by a CLOSE instruction, as described in the following section.

The instruction may be used with Wait or No Wait, as required by the application.

No parameter information is required for this instruction.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	DISCNCT successful
2	Error condition

If successful, the DLS is considered to be in the Disconnected mode, i.e. the only valid instructions that can be issued are CONNECT, CLOSE and READ STATUS.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| DISCNCT |

Example of the DISCNCT instruction

```
        DISCNCT  DSLIN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

| DISCNCT |

2.7 Closing the Communication Link

The CLOSE .DC statement is used to perform the closing of the communication link, when no further transmission is to take place, as follows:

After opening the communication line, and performing the logical connection, as described previously, messages may be sent or received. When all transmission has taken place successfully, and the application no longer requires to send or receive messages, the line must be logically disconnected, as described previously, and then closed by means of this instruction.

The instruction may be used with Wait or No Wait, as required by the application.

No parameter information is required for this instruction.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	CLOSE successful
2	Error condition

If successful, the DLS is considered to be Inactive, until an OPEN instruction is used to re-establish the physical connection.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

Example of the CLOSE instruction

```
        CLOSE .DC,DSLIN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

| CLOSE .DC |

2.8 Data Set Control Instructions

As well as sending and receiving messages, the DSC instruction may be used to handle status and statistic information concerning the line, and for setting timeout values, as follows:

- Set status (DSC Control code `01´) enables the application to inform the system as to whether the connection is switched or leased line.
- Read status (DSC Control code `07´) enables the application to fetch the current state and type of connection.
- Read/reset statistics (DSC Control code `17´) enables the application to fetch statistical information from the software, providing the option was chosen during Monitor generation to include this function.
- Set timeout (DSC Control code `0B´) enables the application to request that it is informed if a connection or receive request is not completed within a pre-determined time limit.

These instructions are explained on the following pages.

2.8.1 Set status

The DSC instruction to Set Status is used to inform the system what type of connection is to be handled, i.e. whether the line is a switched or leased line. If the application does not issue this instruction, the system software assumes the line to be leased, so it is not necessary to use the instruction unless the line is switched. However, if the instruction is used, it must be issued before the OPEN instruction described in section 2.2.

The instruction may be issued with Wait or No Wait, as required by the application. However, if No Wait is used, a WAIT must be issued before the OPEN instruction.

The Control Code for Set Status is X'01', and bit 15 in the binary data item must be set to 0 to indicate Leased line, or 1 to indicate Switched line. Bit 14 may also be set to 0 for 2780 simulation, or 1 for 3780 simulation, assuming the BSC protocol is being used.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	Successful
2	Error condition

The only error condition that can arise is that the instruction is issued when the DLS is not in Inactive mode, i.e. an OPEN instruction has already been successfully completed (and possibly further instructions).

Example of the DSC instruction for Set Status

```
SSTAT   EQU   X'01'
        DSC   DSLIN,SSTAT,LINEC
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

LINEC This is a binary data item, in which bit 15 is set to 0 or 1, to indicate the type of line connection, and bit 14 is set to 0 or 1, to indicate 2780 or 3780 simulation respectively.

2.8.2 Read status

The DSC instruction to Read Status is used to fetch information as to the type of connection (i.e. leased or switched), together with the status of the connection. The instruction may be issued at any time, and may be issued with Wait or No Wait, as required by the application.

The Control Code for Read Status is X'07', and the binary data item will have bits set to indicate the connection and type, as follows:
Bits 14 and 15 indicate the status, and may be:
00 = DLS in Inactive status.
10 = DLS in Active status, but modem not operable.
11 = DLS in Active status, and modem operable (Physical connection has been established, for Switched line).

Bit 11 may be set to:
0 = Data link down
1 = Data link up

Bit 7 may be set to:
0 = Leased connection
1 = Switched connection

Example of the DSC instruction for Read Status

```
RSTAT   EQU   X'07'
        DSC   DSLIN,RSTAT,LINEC
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

LINEC This is a binary data item, which, on completion, will have bits set to indicate the status and type of connection.

2.8.3 Read/Reset Statistics

The DSC instruction to Read / Reset Statistics enables the application to fetch information from the statistics counters in the software. If required, it must be specified during Monitor generation. Optionally, the application may also request that the statistic counters are to be cleared (set to zero) with this instruction.

The instruction may be issued with Wait or No Wait, as required by the application.

The Control Code for Read/Reset Statistics is X`17', and three data items are required to indicate whether the counters are to be cleared, the string into which the information is to be read and the length of the string. The layout of the counters, as they appear in this data item, is shown in the appropriate chapter for the driver used, in module M15A, DC Drivers Reference Manual.

If the counters are to be cleared, the first binary data item must contain the value X`4300', else zeroes.

The string data item must be long enough to contain all the counters that are required by the application, up to a maximum of 62 bytes.

The second binary data item must contain the length in bytes of the information to be read, i.e. if only the first two counters are required, this should contain a value of 4, being two bytes for each counter. In that case, if the option to reset the counters is also set, only the first two counters will be reset.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	Successful
2	Error condition

The only error condition that can arise is that the option to include the statistic counters was not specified during Monitor generation.

Example of the DSC instruction for Read & Reset Statistics

```
STATS   EQU   X'17'
        DSC   DSLIN,STATS,STOPT,CBUF,CLEN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

STOPT This is a binary data item, containing the value X'4300' to indicate that the statistic counters are to be cleared.

CBUF This is a string data item, into which the statistic counter information will be read.

CLEN This is a binary data item, containing the length of the data from the statistic counters that is required, in bytes.

2.8.4 Set Timeout

The DSC instruction to Set Timeout enables the application to set a timer before issuing a CONNECT .PAS, CONNECT .ACT or RECEIVE instruction. This means that, if the request is not successfully completed by the time the timer expires (timeout), the application will be informed, and may take appropriate action.

Note that this instruction must be issued with Wait; the No Wait option is not allowed.

The Control Code for Set Timeout is X'0B', and one binary data item is required, containing the timeout value in 100 milliseconds.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	Successful
2	Error condition

Example of the DSC instruction for Set Timeout prior to RECEIVE

```
STIMO   EQU     X'0B'
        DSC     DSLIN,STIMO,TIMER
        RECEIVE DSLIN,MESLEN,MESBUF,OPTNS
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

TIMER This is a binary data item, containing the timeout value in units of 100 milliseconds.

Chapter 3
MULTIPOINT

3.1 Introduction

A multipoint connection normally consists of the following:

The PTS system is connected, via a modem, to a telephone line (normally leased, but may be switched) and via another modem to another computer at the other end of the line. As far as the CREDIT programmer is concerned, it is not important what type of computer is connected at the other end; it may be another PTS system, or a large mainframe.

In addition, the Monitor software (driver) is designed to look after the protocol that is being used to communicate with the other system. This means that the CREDIT application only has to set up messages that it wishes to transmit, for example, and the driver will ensure that the correct control characters are added, and that the message is acknowledged by the other system, and so forth.

However, the difference between point-to-point connections and multipoint is that in the latter case each task may identify itself to the other system, by means of the sub-addressing feature, and may send and receive messages independently of the other tasks. In effect, the tasks appear to have their own DC line, although in reality they are of course sharing one line.

Note that this does not always apply; it is possible with some drivers to choose not to use the sub-addressing feature. In that case one task identifies the entire DTE to the system at the other end of the line, and any messages sent by any task will be identified by the DTE-address only. Any messages sent to the DTE will be passed to whichever task issued the least recent read request. If no read request is outstanding, the message will be passed to the task that next issues a read request.

It is important to realise that in the first situation, using sub-addresses, it is possible that `spontaneous messages´ are received. A spontaneous message is a message that contains an incorrect sub-address, or that is addressed to a task that does not currently have a read request outstanding. In this case, different options may be available as follows:

- The Monitor may be generated to exclude `message passing´. This means that any message received in which the DTE address is recognized, but the sub-terminal address is not known, and any message that is received when no task has an outstanding read request, will be discarded.

- The Monitor may be generated to include `message passing´. In this case one of the tasks in the application may identify itself as being the task to receive spontaneous messages. The task can identify the sub-address to which the message was addressed by means of one of the data items used in the read request. It is possible for this task to logically disconnect itself from the line, and for another task to then take over the handling of spontaneous messages, by connecting itself and identifying itself as the (new) spontaneous message handler. However, there may only be one task at a time that is assigned to handle such messages.

The Connect Active instruction is normally not used in multipoint connections, where the PTS computer is a secondary station. The general sequence of events that must be controlled by the application are as follows:

Opening the physical connection (by one of the tasks)
Establishing the logical connection
Transmitting messages (after a Poll by the primary station)
Receiving messages
Closing the logical connection
Closing the physical connection (by one of the tasks, if no longer required).

Any task may perform the opening of the communication, and then each task that is to use it must issue a connection request before it can send or receive messages, whether or not the sub-addressing feature is being used.

The application may set a timeout value for CONNECT and RECEIVEoperations requested, by means of the DSC instruction with Control code X`0B´, in the same way as described in Section 2.8.4.

In addition, the DSC instruction can be used for handling status and statistical information, as described in section 3.8.

In the following sections, the CREDIT statements for carrying out the communication over the line are described, with the exception of that for setting timeout.

3.2 Opening the Communication Link

The OPEN .DC instruction is used to perform the physical connection of the computer to the communication line, since after IPL the DTE is considered to be inactive until this request is succesfully completed.

The OPEN instruction may be issued by any of the application tasks, and it is the responsibility of the application to ensure that the OPEN has been successfully completed before any of the tasks issue a CONNECT instruction. This is normally achieved by the setting and clearing of a boolean switch.

The instruction may be issued with Wait or No Wait, as required by the application.

One parameter is required for this instruction, to specify the local DTE-address, as defined during the system design.

The Condition Register will be set to one of the following values as a result of this instruction:

CR Value	Meaning
0	OPEN successful
2	Error condition

If the request is successful, the DTE is considered to be in Disconnected mode (i.e. Active and physically connected, but not logically ready for transmission). Before any messages can be sent or received, the line must be logically connected, as described in the following section.

If an error occurs, the status word can be fetched by means of the XSTAT instruction. The meanings of the bits in the status word are given in the chapter for the appropriate driver in module M15A, Data Communication Driver Reference Manual.

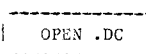
| OPEN .DC |

Example of the OPEN instruction

```
OPEN  .DC,DSLIN,DTEAD
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DTEAD This is a binary data item, containing the local DTE-address. The adddress must be contained in the right byte and the left byte must be set to null (X'00').



3.3 Logically Connecting the Line

The CONNECT .PAS statement is used to perform the logical connection of the DC line, as follows:

After opening the communication line, it is the responsibility of all the tasks that are to use the line to perform a logical connection, to inform the system at the other end of the line of their sub-addresses. In the case where sub-addressing is not used, the tasks must still each issue a logical connection request, although in this case the sub-address specified will not be significant.

The CONNECT .PAS (passive) instruction is used to perform the logical connection of the tasks to the DTE.

The instruction may be used with Wait or No Wait, as required by the application.

Four parameters are required, to specify the local DTE-address, the sub-address and its length, and the option to indicate whether this task is to receive spontaneous messages.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	CONNECT successful
2	Error condition

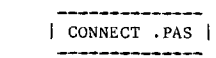
In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

Example of the CONNECT .PAS instruction

```
CONNECT .PAS,DSLIN,DTEAD,DUM,DISCOP,SUBAD,SUBL
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DTEAD This is a binary data item, containing the local DTE-address in the rightmost byte, and X'00' in the lefmost byte.



DUM This is a binary data item, included for syntax reasons. Its contents are not significant.

DISCOP This is a binary data item, containing the value X'00' or X'40' to define whether this task is to receive spontaneous messages. A value of X'40' indicates that this task is to receive such messages.

SUBAD This is a string data item, two bytes long, containing the sub-address associated with this task in the second eight bits, and the first eight bits set to X'00'. In the case where the sub-addressing feature is not included, the contents of this data item are not significant.

SUBL This is a binary data item, containing the length of the string which contains the sub-address, in bytes (X'0002').

| CONNECT .PAS |

3.4 Sending a Message

To send a message on the line, after performing the correct sequence of OPEN and CONNECT .PAS instructions, the SEND instruction is used.

The SEND instruction requires three parameters, to specify the data item containing the message, the length of the message, and (optionally) an indicator to discard pending messages, as follows:

If a message has been received for this task that has not yet been read by means of the RECEIVE instruction, this fact is reported by the setting of bit 7 in the return status. Note, however, that this does not affect the contents of the Condition Register at completion of the request. If the option is specified to discard pending messages, then this bit will not be set, and any outstanding messages addressed to this task will be discarded before execution of the instruction.

The instruction may be issued with or without Wait, as required by the application.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	SEND successful
2	Error condition

In the case of the Condition Register being set to zero, it is stilladvisable to fetch the status word via the XSTAT instruction, since, if bit 7 is set, this indicates that the SEND has been successfully completed, but that a message has been received for this task, and therefore a RECEIVE should be issued as soon as possible to collect the message. This does not apply if the option is set to discard pending received messages, as described previously.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| SEND |

Example of the SEND instruction

```
SEND  .NW,DSLIN,MESBUF,MESLEN,OPTNS
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

MESBUF This is a string data item, containing the message to be sent on the line. The driver will add the necessary framing characters before transmission.

MESLEN This is a binary data item, containing the length of the message to be sent, in number of characters. Thus the string data item may be longer than the actual message, as the number of characters sent is determined by the value in this data item.

OPTNS This is a binary data item containing the value X`0020' to indicate that pending messages are to be discarded (otherwise X`0000').

| SEND |

3.5 Receiving a Message

To receive a message on the line, after performing the correct sequence of OPEN and CONNECT .PAS instructions, the RECEIVE instruction is used.

The RECEIVE instruction requires three parameters, to specify the data item to contain the message, the length of this data item, and, in the case of refused requests or spontaneous messages, to receive the status or sub-address respectively.

The instruction may be issued with or without Wait, as required by the application.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	RECEIVE successful
2	Error condition

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

In the case of bit 7 being set in the status word, this indicates that at least one further message is waiting for this task, and another RECEIVE must be issued as soon as possible, in order to release the buffer. Note that, if only this bit is set, the Condition Register will be set to zero, thus it is recommended to issue the XSTAT instruction in any case.

Example of the RECEIVE instruction

```
RECEIVE .NW,DSLIN,MESBUF,MESLEN,STAT
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

MESBUF This is a string data item, into which the message will be read from the line. The driver will remove any framing characters before placing the message in this data item.

MESLEN This is a binary data item, containing the maximum expected length of the message to be read. Thus the string data item may be longer than the actual message, as the number of characters read is determined by the value in this data item. If the length is shorter than the actual message, an error will be reported via the Condition Register.

STAT This is a binary data item, which, in the case of an error, will contain the status of the connection, as defined in section 3.8.1, or the sub-address to which the message is addressed, in the case of spontaneous message reception.

| RECEIVE |

3.6 Logically Disconnecting the Line

The DISCNCT statement is used to perform the logical disconnection of the DC line, as follows:

After opening the communication line, and performing the logical connection, as described previously, a task may send and receive messages. When transmission has been completed (i.e. the task no longer wishes to use the line), disconnection must take place. Any task that has performed a logical connection may issue a disconnect request, and this must normally be carried out before the CLOSE instruction can be issued to physically disconnect the line, as described in the following section. The DISCNCT instruction can also be used after a timeout on a RECEIVE request, to avoid receiving spontaneous messages.

If the sub-addressing feature is not included, then only the task that performed the Connect needs to issue the Disconnect request.

The instruction may be used with Wait or No Wait, as required by the application.

One parameter is required for this instruction, being the symbolic local sub-address, unless the sub-addressing feature is not included.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	DISCNCT successful
2	Error condition

If successful, the DTE is considered to be in the Disconnected mode, i.e. the only valid instructions that can be issued by this task are a Connect, Close, or Read Status.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| DISCNCT |

Example of the DISCNCT instruction

```
DISCNCT  DSLIN,SUBAD
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

SUBAD This is a string data item, two bytes long, containing the sub-address associated with this task in the second byte, and the first byte set to X`00´. In the case where the sub-addressing feature is not included, this data item is not required.

| DISCNCT |

3.7 Closing the Communication Link

The CLOSE .DC statement is used to perform the closing of the communication link, when no further transmission is to take place, as follows:

After opening the communication line, and performing the logical connection, messages may be sent or received. When all transmission has taken place successfully, and the application no longer requires to send or receive messages, the line must be logically disconnected, as described previously, and then closed by means of this instruction.

The instruction may be used with Wait or No Wait, as required by the application.

An optional parameter may be supplied, to indicate whether this request is to be Conditional or Unconditional, as follows:
- A conditional Close request is only executed if no tasks are logically connected (i.e. all tasks in a system with the sub-addressing feature have successfully issued a Disconnect request.
- An unconditional Close request is always executed; thus if there are still messages waiting in queues, these are discarded, and no task may now use the line.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	CLOSE successful
2	Error condition

If successful, the DTE is considered to be Inactive, until an OPEN instruction is used to re-establish the physical connection.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| CLOSE .DC |

Example of the CLOSE instruction

```
CLOSE .DC,DSLIN,DUM,CLOPT
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DUM This is a binary data item, included for syntax purposes; its contents are not significant.

CLOPT This is a binary data item, containing the value X`0080' if this is to be a Conditional Close, or X`0000' if it is to be Unconditional.

```
-------------
| CLOSE .DC |
-------------
```

3.8 Data Set Control Instructions

As well as sending and receiving messages, the DSC instruction may be used to handle status and statistic information concerning the line, as follows:

- Read status (DSC Control code `07') enables the application to fetch the current state and type of connection.
- Read/reset statistics (DSC Control code `17') enables the application to fetch statistical information from the software, providing the function was included during Monitor generation

The use of these instructions is described on the following pages.

3.8.1 Read Status

The DSC instruction to Read Status is used to fetch information as to the status of the connection. The instruction may be issued at any time, and may be issued with Wait or No Wait, as required by the application.

The Control Code for Set Status is X'07', and the binary data item will have bits set to indicate the connection and type, as follows:

Bit 15 indicates the status, and may be:
0 = DTE-DCE interface is Active.
1 = DTE-DCE interface is Inactive.

Bit 11 may be set to:
0 = Data link down
1 = Data link up

Bits 6 & 7 may be set to:
00 = DTE Closed
01 = OPEN in progress
10 = DTE Open
11 = CLOSE in progress

Bits 3 & 4 may be set to:
00 = Task is Disconnected
01 = CONNECT in progress
10 = Task is Connected
11 = DISCONNECT in progress

Example of the DSC instruction for Read Status

```
RSTAT   EQU   X'07'
        DSC   DSLIN,RSTAT,LINEC
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

LINEC This is a binary data item, which, on completion, will have bits set to indicate the status of the connection.

3.8.3 Read/Reset Statistics

The DSC instruction to Read / Reset Statistics enables the application to fetch information from the statistics counters in the software. If required, it must be specified during Monitor generation. Optionally, the application may also request that the statistic counters are to be cleared (set to zero) with this instruction.

The instruction may be issued with Wait or No Wait, as required by the application.

The Control Code for Read/Reset Statistics is X'17', and three data items are required to determine whether the counters are to be cleared, the string into which the information is to be read and the length of the string. The layout of the counters, as they appear in this data item, is shown in the appropriate chapter for the driver used, in module M15A, DC Drivers Reference Manual.

If the counters are to be cleared, the first binary data item must contain the value X'4300', else zeroes.

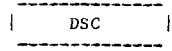
The string data item must be long enough to contain all the counters that are required by the application, up to a maximum of 62 bytes.

The second binary data item must contain the length in bytes of the information to be read, i.e. if only the first two counters are required, this should contain a value of 4, being two bytes for each counter. In that case, if the option to reset the counters is also set, only the first two counters will be reset.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	Successful
2	Error condition

The only error condition that can arise is that the option to include the statistic counters was not specified during Monitor generation.



Example of the DSC instruction for Read & Reset Statistics

```
STATS   EQU   X'17'
        DSC   DSLIN,STATS,STOPT,CBUF,CLEN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

STOPT This is a binary data item, containing the value X'4300' to indicate that the statistic counters are to be cleared.

CBUF This is a string data item, into which the statistic counter information will be read.

CLEN This is a binary data item, containing the length of the data from the statistic counters that is required, in bytes.

| DSC |

Chapter 4

NETWORKS

4.1 Introduction

A network connection normally consists of the following:

The PTS system is connected, via a modem, to a DC network and, via the network, to another computer (the remote DTE) also connected to the network. As far as the CREDIT programmer is concerned, it is not important which network is used and which type of computer is connected at the other end; it may be another PTS system, or a large mainframe.

In addition, the Monitor software (driver) is designed to look after the protocol that is being used by the network. This means that the CREDIT application has to set up the connection with the remote DTE with which it wants to communicate, and to prepare messages that it wishes to transmit, for example. The driver will ensure that the correct control characters for transport by the network are added, and that the message is acknowledged by the other system, and so forth.

In network connections, each DTE is known to the network by its network address, and the application by a symbolic DTE address. The relation between network address and symbolic address is defined in the configuration data (see chapter 5). If subaddresses are used to specify a task or terminal device, these must be handled by the application.

In a network connection, the Connect Active request can be used to establish a logical connection and the DTEs with which the connection must be made, can be specified in the request.

In error situations, network information may be obtained and may have to be acted upon by the application.

The general sequence of events that must be controlled by the application are as follows:

Opening the physical connection (by one of the tasks)
Establishing the logical connection
Transmitting messages and/or
Receiving messages
Closing the logical connection
Closing the physical connection (by one of the tasks, if no longer required).

Any task may open the physical connection, and then each task that is to use it must issue a connection request to establish the logical connection before it can send or receive messages. Each task that has established a logical connection must issue a disconnection request before the physical connection can be closed.

There are two instructions for establishing the logical connection: Connect Active and Connect Passive. Connect Active is used to notify the network that the task wants to establish a logical connection with the specified DTE. Connect Passive is used to notify the network that the task is ready to make a logical connection with a DTE that issues a Connect Active request.

In addition, the DSC instruction can be used for handling status and statistical information, as described in section 4.8.

In the following sections, the CREDIT statements for carrying out the communication over the line are described, with the exception of that for setting timeout.

The application may set a timeout value for the Open, Close, Connect, Send and Receive operations requested, by means of the DSC instruction with Control code X'0B', as described in Section 2.8.4.

4.2 Opening the Communication Link

The OPEN .DC instruction is used to perform the physical connection of the computer to the communication line, since after IPL the DTE is considered to be inactive until this request is successfully completed.

The OPEN instruction may be issued by any of the application tasks, and it is the responsibility of the application to ensure that the OPEN has been successfully completed before any of the tasks issue a CONNECT instruction. CONNECT instructions issued by application tasks to set up a logical connection, will not be successful before an OPEN instruction has been successfully completed.

The instruction may be issued with Wait or No Wait, as required by the application.

One parameter is required for this instruction, to specify the local DTE-address as defined during the system design.

Optionally a table with network addresses may be supplied, which will then replace the network addresses supplied in the configuration data. This is described in the chapter for the appropriate driver in module M15A, Data Communication Driver Reference Manual.

The Condition Register will be set to one of the following values as a result of the Open instruction:

CR Value	Meaning
0	OPEN successful
2	Error condition

If the request is successful, the DTE is considered to be in Disconnected mode (i.e. Active and physically connected, but not logically ready for transmission). Before any messages can be sent or received, the logical connection must be made between the DTE and the DWT with a Connect instruction as described in the following section.

If an error occurs, the status word can be fetched by means of the XSTAT instruction. The meanings of the bits in the status word are given in the chapter for the appropriate driver in module M15A, Data Communication Driver Reference Manual.

| OPEN .DC |

Example of the OPEN instruction

```
OPEN  .DC,DSLIN,DTEAD,BUF,LENG
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DTEAD This is a binary data item, containing the local DTE-address. The address must be supplied as the hexadecimal representation of two ASCII characters.

BUF Address of the string containing new network addresses. (Optional).

LENG Binary data item containing the length of string containing the new network addresses in bytes. (Optional).

| OPEN .DC |

4.3 Logically Connecting the Line

4.3.1 Connect Passive

The Connect Passive instruction is used to make the logical connection between the task that issues the request, and the local DTE. In addition it informs the driver that the task is ready to set up a logical connection, and that a Connect Active request from a remote DTE can be accepted.

The instruction may be used with Wait or No Wait, as required by the application.

One parameter is required, the local symbolic DTE address. As an option, it can be specified that a connection can be made with any system connected to the network, or the symbolic DTE address can be supplied of one DTE that is allowed to make a connection.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	CONNECT successful
2	Error condition

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| CONNECT .PAS |

Example of the CONNECT .PAS instruction

```
CONNECT .PAS,DSLIN,DTEAD,DTERAD,CPOPT
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DTEAD This is a binary data item, containing the local symbolic DTE-address as the hexadecimal representation of two ASCII characters.

DTERAD This is a binary data item, containing the remote symbolic DTE-address as the hexadecimal representation of two ASCII characters. If no remote symbolic DTE address must be supplied but there are options to be specified, DTERAD must be included for syntax reasons as a dummy item, containing hexadecimal zeroes (X`0000´).

CPOPT This is a binary data item, containing the value X`00´ or the value for driver dependent options.

| CONNECT .PAS |

4.3.2 Connect Active

The Connect Active instruction is used to make the logical connection between the task that issues the request, and the local DTE. In addition it informs the driver that the task wants to set up a logical connection with a remote DTE, which must have issued a Connect Passive request.

The instruction may be used with Wait or No Wait, as required by the application.

Two parameters are required, the local and the remote symbolic DTE address. Driver dependent options can be included in the instruction parameters.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	CONNECT successful
2	Error condition

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

Example of the CONNECT .ACT instruction

```
CONNECT .ACT,DSLIN,DTEAD,DTEREM,CAOPT
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DTEAD This is a binary data item, containing the local symbolic DTE-address as the hexadecimal representation of two ASCII characters.

DTEREM This is a binary data item, containing a remote symbolic DTE address as the hexadecimal representation of two ASCII characters. If no remote symbolic DTE address must be supplied but there are options to be specified, DTERAD must be included for syntax reasons as a dummy item, containing hexadecimal zeroes (X'0000').

CAOPT This is a binary data item, containing the value X'00' or the value for driver dependent options.

| CONNECT .ACT |

4.4 Sending a Message

To send a message via the network, after performing the correct sequence of OPEN and CONNECT .PAS or .ACT instructions, the SEND instruction is used.

The SEND instruction requires two parameters specifying the data item containing the message and the length of the message. Driver dependent options may be set in addition to these parameters.

The Send Addressed instruction is used if the network protocol, the network driver or the application require that the local and/or remote symbolic DTE address is specified in the instruction. In that case the keyword .ADDR indicates that these items are included in the parameter list.

The instruction may be issued with or without Wait, as required by the application.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	SEND successful
2	Error condition

In the case of the Condition Register being set to zero, it is stilladvisable to fetch the status word via the XSTAT instruction, since, if bit 7 is set, this indicates that the SEND has been successfully completed, but that a message has been received for this task, and therefore a RECEIVE should be issued as soon as possible to collect the message. This does not apply for all network drivers, see module M15A, Data Communication Driver Reference Manual.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

Example of the SEND instruction

```
        SEND  .NW,DSLIN,MESBUF,MESLEN,OPTN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

SEND

MESBUF This is a string data item, containing the message to be sent on the line. The driver will add the necessary framing characters before transmission.

MESLEN This is a binary data item, containing the length of the message to be sent in bytes. Thus the string data item may be longer than the actual message, as the number of characters sent is determined by the value in this data item.

OPTNS This is a binary data item containing the value for driver dependent options.

| SEND |

4.5 Receiving a Message

To receive a message on the line, after performing the correct sequence of OPEN and CONNECT .PAS instructions, the RECEIVE instruction is used.

The RECEIVE instruction requires two parameters, specifying the data item to contain the message and the length of the message.

The instruction may be issued with or without Wait, as required by the application.

The Receive Addressed instruction is used if the network protocol, the network driver or the application require that the local and/or remote symbolic DTE address is specified in the instruction. In that case the keyword .ADDR indicates that these items are included in the parameter list.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	RECEIVE successful
2	Error condition

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

In the case of bit 7 being set in the status word, this indicates that at least one further message is waiting for this task, and another RECEIVE must be issued as soon as possible, in order to release the buffer. This does not apply for all drivers, see module M15A, Data Communication Driver Reference Manual. Note that, if only bit 7 is set, the Condition Register will still be zero, therefore it is recommended to fetch the status word in any case.

RECEIVE

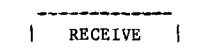
Example of the RECEIVE instruction

```
RECEIVE .NW,DSLIN,MESBUF,MESLEN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

MESBUF This is a string data item, into which the message will be read from the line. The driver will remove any framing characters before placing the message in this data item.

MESLEN This is a binary data item, containing the maximum expected length of the message to be read, in bytes. Thus the string data item may be longer than the actual message, as the number of characters read is determined by the value in this data item. If the length is shorter than the actual message, an error will be reported via the Condition Register.



4.6 Logically Disconnecting the Line

The DISCNCT instruction is used to terminate the logical connection of the local and remote DTE.

After opening the communication line and performing the logical connection, a task may send and receive messages. Any task that has performed a logical connection must issue a disconnect request when transmission has been completed (i.e. the task no longer wishes to use the line). Disconnection must normally take place before the CLOSE instruction can be issued to physically disconnect the line, as described in section 4.7.

The Disconnect request is often used after a Receive has been completed with Timeout, to avoid receiving spontaneous messages. The sequence of instructions is in that case:
CONNECT - SEND - RECEIVE (Timeout) - DISCONNECT - CONNECT - etc.
The instruction may be used with Wait or No Wait, as required by the application.

Parameters required for this instruction are driver dependent, see module M15A, Data Communication Driver Reference Manual.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	DISCNCT successful
2	Error condition

If successful, the DTE is considered to be in the Disconnected mode, i.e. the only valid instructions that can be issued by this task are a Connect Active or Passive, Close, or Read Status.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

Example of the DISCNCT instruction

```
        DISCNCT  DSLIN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DISCNCT

4.7 Closing the Communication Link

The CLOSE .DC instruction is used to perform the closing of the communication link, when no further transmission is to take place, as follows:

When all transmission has taken place successfully, and the application no longer requires to send or receive messages, the line must be logically disconnected, as described previously, and then physically closed by means of the CLOSE instruction.

The instruction may be used with Wait or No Wait, as required by the application.

The local symbolic DTE address is a parameter required for some drivers and optional for others.

For some DC drivers an optional parameter may be supplied, to indicate whether this request is to be Conditional or Unconditional, as follows:

- A Conditional Close request is only executed if no tasks are logically connected (i.e. all tasks that have issued a Connect Active or Passive have successfully issued a Disconnect request).

- An Unconditional Close request is always executed; thus if there are logical connections still active, the requests are completed with the value in the Status Word indicating "Physical Connection Error".

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	CLOSE successful
2	Error condition

If successful, the DTE is considered to be Inactive, until an OPEN instruction is used to re-establish the physical connection.

In the case of an error, the status word can be fetched from the driver with the XSTAT instruction. The meanings of the bits in this word are given in the chapter for the driver concerned in module M15A, Data Communication Driver Reference Manual.

| CLOSE .DC |

Example of the CLOSE instruction

```
CLOSE .DC,DSLIN,DTEAD,CLOPT
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

DTEAD This is a binary data item containing the local symbolic DTE address as the hexadecimal representation of two ASCII characters.

CLOPT This is a binary data item, containing a driver dependent value indicating Conditional or Unconditional Close.

```
| CLOSE .DC |
```

4.8 Data Set Control Instructions

As well as sending and receiving messages, two further instructions may be used to handle status and statistic information concerning the line, as follows:

- Read status (DSC Control code `07') enables the application to fetch the current state and type of connection.
- Read/reset statistics (DSC Control code `17') enables the application to fetch statistical information from the software if the function was included in the driver during Monitor generation.

These instructions rae described in the following pages.

4.8.1 Read status

The DSC instruction to Read Status is used to fetch information as to the status of the connection. The instruction may be issued at any time, and may be issued with Wait or No Wait, as required by the application.
Depending on the driver used, additional network information may be obtained in the buffer specified in the instruction.

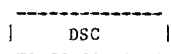
The Control Code for Read Status is X'07', and the binary data item will have bits set to indicate the connection and type, as specified for the driver used and described in module M15A, DC Drivers Reference Manual.

Example of the DSC instruction for Read Status

```
RSTAT   EQU   X'07'
        DSC   DSLIN,RSTAT,LINEC
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

LINEC This is a binary data item, which, on completion, will have bits set to indicate the status of the connection.



4.8.2 Read/Reset Statistics

The DSC instruction to Read/Reset Statistics enables the application to fetch information from the statistics counters in the software. If required, it must be specified during Monitor generation. Optionally, the application may also request that the statistic counters are to be cleared (set to zero) with this instruction, but this option is not available in all DC drivers.

The instruction may be issued with Wait or No Wait, as required by the application.

The Control Code for Read/Reset Statistics is X'17', and three data items are required to determine whether the counters are to be cleared, the string into which the information is to be read and the length of the string. The layout of the counters, as they appear in this data item, is shown in the appropriate chapter for the driver used, in module M15A, DC Drivers Reference Manual.

If the counters are to be cleared, the first binary data item must contain the value X'4300', else zeroes.

The string data item must be long enough to contain all the counters that are required by the application, up to a maximum of 62 bytes.

The second binary data item must contain the length in bytes of the information to be read, i.e. if only the first two counters are required, this should contain a value of 4, being two bytes for each counter.

Note, however, that a request to reset the counters, when the specified length is less than the total statistic area, will only reset the counters that are read.

The Condition Register will be set as a result of this instruction to one of the following values:

CR Value	Meaning
0	Successful
2	Error condition

The only error condition that can arise is that the option to include the statistic counters was not specified during Monitor generation.

DSC

Example of the DSC instruction for Read & Reset Statistics

```
STATS   EQU   X'17'
        DSC   DSLIN,STATS,STOPT,CBUF,CLEN
```

DSLIN This is the data set identifier for the DC line. The TOSS file code must have been defined at Monitor generation, and must have been defined in the application for use in this terminal class.

STOPT This is a binary data item, containing the value X'4300' to indicate that the statistic counters are to be cleared.

CBUF This is a string data item, into which the statistic counter information will be read.

CLEN This is a binary data item, containing the length of the data from the statistic counters that is required.

| DSC |