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.

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.

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.

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.

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.

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.

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.

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.