Mitel Technical Specification 22

MiNET Specification

Neil Sipkes
Technical Requirements and Standards

Version B05 (Stephen Lee)
4th October 2001

Reviewers for MTS22

This is to acknowledge the people who reviewed version A01 of this document:

Roland Michaud	Peripheral Firmware
Gary Lam	NTDD
Bob Armstrong	Product Verification
Steve Duhn	Product Test Automation
Dave Perry	Peripheral Firmware
John Boyd	GX5000
Sue Welsh	Technical Trials & Traffic Performance

The conformance testing (section D) was written by Steve Duhn.

This is to acknowledge the people who provided input or reviewed subsequent versions:

Roland Michaud	Communicating Objects
Dave McNeil	Migration & Peripherals Software
Pierre Lavallée	Migration & Peripherals Software
Chris Nason	Migration & Peripherals Software
Kim Currie	System Control & Digital Terminals
Radovan Prodanovic	System Control & Digital Terminals
Roger Bastin	System Control & Digital Terminals
Normand Cyr	Network Enhancement Products
Tom McGuinness	Peripherals and Interfaces
Dave Perry	System Control & Digital Terminals
Steve Duhn	Automated Testing Solution
Andy Weatherson	Network Enhancement Products
François Audet	Systems Design and Standards
Scott Walker	Automated Testing Solution
Carl Heyendal	System Control & Digital Terminals
Robin Rock	System Control & Digital Terminals

Отформатировано: французский (Франция)

Код поля изменен

Отформатировано: французский (Франция)

Код поля изменен

Отформатировано: французский (Франция)

Note: *Library copies of the MTS can be signed out from the Technical Requirements and Standards library for those employees who need only to reference the document for a short period of time.*

Document History

Version	Date	Description	Author (s)
A01	06-NOV-1992	First release of MTS22	Neil Sipkes
A02	30-APR-1993	Added new cadence setting procedures	François Audet
A03	06-JUL-1993	Added “deactivate prompt” to display prompt	François Audet
A04	20-SEP-1993	Added new message (23) for cadence setting	François Audet
A05	28-OCT-1993	Modification of the set ID codes	François Audet
A06	11-MAR-1994	Added new messages D3, D4, 05	François Audet
A07	21-JUN-1994	Added Annex A to section C	François Audet
B01	23-AUG-1995	Added Section D, messages D5, D6 & set ID 90 and set filter coefficient in config. h/w command	Steve Duhn & François Audet
B02	30-OCT-1995	Corrections to sections A, C and D	François Audet
B03	31-OCT-1996	Added set IDs for 600 series Changes to “configure hardware” codepoints Corrections to section D	Dave Walker & Scott Walker
B04	01-NOV-1997	Added new messages and fields for 4000 series sets	Dave Walker
B05	18-June-1999	Converted document from frame to word 7 format	Kelvin Steeden
B06	10-Aug-2001	General update	Stephen Lee

Introduction

Mitel Technical Specifications for DNIC, MiLINK, MiLAP, and MiNET

Mitel Technical Specifications (MTS) for DNIC, MiLINK, MiLAP, and MiNET define the physical, electrical, and procedural requirements (corresponding to Layers 1, 2, and 3 of the OSI Reference Model) for digital sets and peripherals. In addition, the scope of the specifications for MiLAP and MiNET extends to the system level as well.

The purpose of the specifications is to ensure correct functioning and consistency of implementation. These MTSs are based on internal documents generated by various development groups, and also reflect the latest status of the Protocols Working Group (PWG). Relevant test principles are included wherever possible to verify conformance to each specification. Wherever appropriate, safety or overvoltage requirements, electromagnetic compatibility limitations or other compatibility or regulatory requirements are also included for completeness.

The interpretation of these MTSs is the sole responsibility of the Systems Design and Standards. Periodic updates will be issued to ensure, amongst other things, compliance with the most recent decisions of the PWG.

The Mitel Technical Specifications for DNIC, MiLINK, MiLAP, and MiNET are structured as follows:

MTS20 MiLINK Specification
MTS21 MiLAP/MiLAP-S Specification
MTS22 MiNET Specification
MTS23 DNIC Specification

Section A - Overview

1.0 Document Format of MTS22

This Mitel Technical Specification (MTS22) is divided into four sections:

Section A	Provides an overview of MiNET.
Section B	Provides a general introduction to the concepts of the layer 3 protocol.
Section C	Provides the protocol definition. The information in this section is based on Mitel documents PS.0 [11], PS.1 [12], PS.2 [13], PS.3 [14], PS.4 [15], PS.5 [16], PS.6 [17], PS.8 [18], and TN.0 [19].
Section D	Details the conformance testing to be used in verifying the protocol definition of Section C.

Text in a dashed-line box provides definitive interpretation where the baseline text is unclear or ambiguous.

1.1 Abbreviations and Acronyms Used in MTS22

AIM	Analogue Interface Module
ALM	Analogue Line Monitor
ASCII	American Standard Code for Information Interchange
BLF	Busy Lamp Field
CAU	Call Announce Unit
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CODEC	Coder/decoder (for PCM voice)
CPB	Cadence Pattern Byte
DCE	Data Circuit-terminating Equipment
DLIC	Digital Line Interface Circuit
DMP	DNIC Music on hold/Pager Module
DNIC	Digital Network Interface Circuit
DTE	Data Terminal Equipment
IBM	International Business Machines
ID	Identification or identifier
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU - Telecommunication Standardization Sector (ex-CCITT)
IUT	Implementation Under Test
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
MCX	Mitel “coax” transport protocol
MiLAP	Mitel Link Access Procedure
MiLINK	Mitel proprietary physical interconnect scheme
MiNET	Mitel Network layer protocol
MT	Message Type
MTS	Mitel Technical Specification
OSI	Open System Interconnection (an ISO specification)
PBX	Private branch exchange
PC	Personal Computer
PCM	Pulse Code Modulation
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation Extra Information for Testing

PKM Programmable Key Module
RAM Random Access Memory
ROM Read-Only Memory
SAP Service Access Point
SDU Service Data Unit
SS7 SuperSet 7 console
sP Mitel simplified Protocol (a combined layer 2 and layer 3 protocol)
TTCN Tree and Tabular Combined Notation
TEI Terminal Endpoint Identifier

1.2 References

1. ANSI X3.4, Code for Information Interchange, 1986.
2. CCITT Recommendation V.24, List of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE), 1988.
3. CCITT Recommendation V.35, Data transmission at 48 kilobits per second using 60-108 kHz group band circuits, 1988.
4. CCITT Recommendation X.21, Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for synchronous operation on public data networks, 1988.
5. CCITT Recommendation X.200, Reference Model of Open Systems Interconnection for CCITT Applications.
6. CCITT Recommendation X.210, OSI Layer Service Conventions.
7. CCITT Recommendation X.211, Physical Service Definition of Open Systems Interconnection for CCITT Applications.
8. CCITT Recommendation X.212, Data Link Service Definition of Open Systems Interconnection for CCITT Applications.
9. Mitel MTS20, MiLINK Specification, Version A01
10. Mitel MTS21, MiLAP/MiLAP-S Specification, Version B01
11. Mitel STD PS.0, MiNET/Voice II - Mitel Voice Set and Console Layer 3 Protocol, Version A03, March 1992
12. Mitel STD PS.1, MiNET - Mitel Network Layer Protocol Common Elements, Version A02, March 1992
13. Mitel STD PS.2, MiNET/Async - Async Data Transport Protocol, Version A05, March 1992
14. Mitel STD PS.3, MiNET/Sync - Sync Data Transport Protocol, Version A01, March 1992.
15. Mitel STD PS.4, MiNET/MCX - Mitel Coax Rate Adaption, Version A01, March 1992
16. Mitel STD PS.5, MiNET/DNLD - MiNET Downloading Procedures, Version A01, March 1992
17. Mitel STD PS.6, MDI/SS7 - SS7 Transport Protocol, Version A00, September 1992
18. Mitel STD PS.8, MiNET/DATA - Mitel Data Set Network Layer, Version A01, March 1992.
19. Mitel SETS TN.0, MiNET Voice Set State Machine Description, Version A02, August 1992
20. Mitel SETS TN.2, Summary of Sets Specifications, Radovan Prodanovic, Dave Perry, Version A08, 22 September 1993
21. EIA RS-232 (superseded by TIA-232-E-91).
22. EIA-232-D (superseded by TIA-232-E-91).
23. TIA-232-E-91, Interface between data terminal equipment and data circuit-terminating equipment employing serial binary data interchange, 1991.
24. ISO/IEC 9646, Information technology - Open Systems Interconnection - Conformance testing methodology and framework,
    - Part 1: General concepts, 1991
    - Part 2: Abstract test suite specification, 1991
    - Part 3: The Tree and Tabular Combined Notation (TTCN), 1992
    - Part 4: Test realization, 1991
    - Part 5: Requirements on test laboratories and clients for the conformance assessment process, 1991

Section B - General Concepts

1.0 General

This specification (MTS22-B) describes in general terms the functions of the MiNET protocol. Detailed descriptions of the protocol are contained in MTS22-C.

Although the MiNET protocol is not in strict accordance with the requirements specified in the OSI reference model, the term layer 3 will at times be used in this document.

MiNET provides the means to establish, maintain and terminate call connections between systems (eg. PBXs) and terminals (eg. digital telephone sets, data sets, etc.). In addition, the protocol provides configuration and maintenance capabilities from the system to terminals, as well as rate adaption capabilities for B-channel data connections.

MiNET makes use of the services provided by a suitable data link layer (layer 2). Specifically, the data link layer provides the following functions to layer 3:

a) establishment of data link connections;
b) error-protected transmission of data; and
c) re-establishment of data link connection (indicating loss of information).

Note- *Examples of two suitable data link layer protocols (MiLAP and MiLAP-S) can be found in MTS21.*

Layer 3 makes invisible to the user how it utilizes underlying resources, such as the data link layer, to provide its functions.

MiNET is applicable to both D-channel and B-channel communication.

1.1 Structure of MiNET

There are three categories of services provided by MiNET:

a) call control;
b) terminal configuration and maintenance; and
c) data transport.

Within the category of data transport, the following modes are supported:

a) asynchronous ASCII data at speeds of 19.2 kbit/s or less;
b) synchronous data at speeds of 19.2 kbit/s or less;
c) IBM 3270-type synchronous data (terminal to controller); and
d) bulk data transport (ie. software downloading).

Functions performed by MiNET include the following:

a) processing of primitives for communication with the data link layer entity;
b) processing of primitives for communication with upper layers, when present;
c) generation and interpretation of layer 3 messages for peer-level communication; and
d) administration of call control, data transport, and terminal configuration and maintenance.

1.2 Interface between layer 3 and adjacent layers

When an upper layer is present, layer 3 provides its services via a layer 3 service access point (SAP). Similarly, layer 3 receives services from the data link layer via a data link layer SAP. Refer to Figure 1/MTS22-B. A particular service is provided to the upper layer, or received from the lower layer, via the exchange of sequences of primitives across the SAP. Primitives represent, in an abstract way, the logical exchange of information and control between layers, and do not specify or constrain implementation.

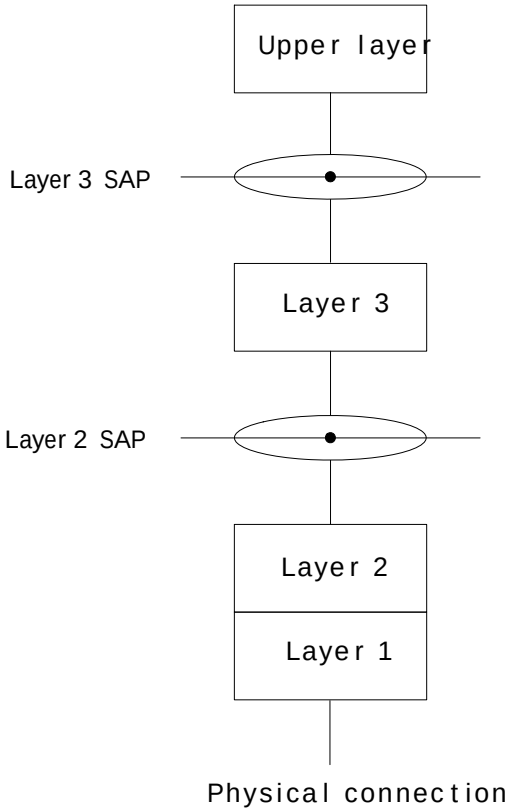


Figure 1/MTS22-B
Interfaces between layer 3 and adjacent layers

Section C - Protocol Definition

1.0 General

This section (MTS22-C) describes the message formats, types and contents for the MiNET protocol. In addition, primitives are defined for use between MiNET and upper protocol layers.

MiNET uses the services of a suitable data link layer protocol (e.g. MiLAP or MiLAP-S). MiNET utilizes these services using primitives defined by the data link layer protocol (see, for example, MTS21 [10]. Such primitives are used to illustrate the communication between protocol layers and are not intended to specify or constrain implementations.

This document will not provide detailed specifications for messages used by the older DLIC-based terminals. For those messages, refer to § 8/PS.0 [11].

Note 1: - *As stated in MTS22-B, although MiNET is not in accordance with the definitions of an OSI layer 3 protocol, the term layer 3 will at times be used in this specification.*

Note 2: - *The term terminal will be used throughout this specification to refer to sets and peripherals. The term system will be used to refer to PBX and set-handler.*

2.0 Elements of layer-to-layer communication

2.1 General

Communications between layers are accomplished by the means of primitives.

Primitives represent, in an abstract way, the logical exchange of information and control between layer 3 and adjacent layers. They neither specify nor constrain implementations.

Primitives consist of commands and their respective responses associated with the services requested of a lower layer. The general syntax of a primitive is:

XX - Generic name - Type: Parameters

where XX designates the interface across which the primitive is active.

For this MTS, the following definition for XX is used:
- DL for communication between layer 3 and the data link layer.

Data link service definitions and primitives are defined in CCITT Recommendation X.212 [8]. It may prove helpful to consult this document for a better understanding of primitives.

2.1.1 Generic Names

The generic name of a primitive specifies the activity that should be performed. Table 1/MTS22-C illustrates the primitives defined in this MTS. Note that not all primitives have associated parameters. The primitive generic names that are defined in this MTS are described in §§ 2.1.1.1 to 2.1.1.6.

2.1.1.1 DL-ESTABLISH

The DL-ESTABLISH primitives are used to request, indicate and confirm the outcome of the procedures for establishing multiple frame operation.

2.1.1.2 DL-RELEASE

The DL-RELEASE primitives are used to request, indicate and confirm the outcome of the procedures for terminating a previously established multiple frame operation, or for reporting an unsuccessful establishment attempt.

2.1.1.3 DL-DATA

The DL-DATA primitives are used to request and indicate service data units (SDUs) containing layer 3 protocol data units (PDUs) which are to be transmitted, or have been received, by the data link layer using the acknowledged information transfer service.
Table 1/MTS22-C

Primitives associated with
MTS22

Generic Name	Type			Parameters		Parameter Data Contents
	Request	Indication	Confirm	Priority indicator	Parameter Data	
L3 ? L2						
DL-ESTABLISH	X	X	X	-	-	
DL-RELEASE	X	X	X	-	-	
DL-DATA	X	X	-	-	X	Layer 3 PDU (peer-to-peer message)
DL-UNIT-DATA	X	X	-	-	X	Layer 3 PDU (peer-to-peer message)
DL-ASSIGN	X	-	-	-	X	Address value
DL-REMOVE	-	X	-	-	-	

L3 ? L2: Layer 3/data link layer boundary

Note - The definition of primitives between MiNET and upper layers is for further study.

2.1.1.4 DL-UNIT-DATA

The DL-UNIT-DATA primitives are used to request and indicate SDUs containing layer 3 PDUs which are to be transmitted, or have been received, by the data link layer using the unacknowledged information transfer service.

2.1.1.5 DL-ASSIGN

The DL-ASSIGN primitive is used by layer 3 to request that the data link layer use the address or TEI value contained within the parameter data of the primitive.

2.1.1.6 DL-REMOVE

The DL-REMOVE primitive is used by the data link layer to notify layer 3 that the current address or TEI should no longer be considered valid. This condition normally occurs as a result of a physical layer discontinuity.

2.1.2 Primitive types

The primitive types defined in this MTS are:

1) the *request* primitive type is used when a higher layer is requesting a service from the lower layer;
2) the *indication* primitive type is used by a layer providing a service to inform the higher layer; and
3) the *confirm* primitive type is used by the layer providing the requested service to confirm that the activity has been completed.

2.1.3 Parameter definition

A parameter consists of data, such as: address or reasons. The parameter data is associated with the primitive and contains information related to the service. In the case of DATA primitives, the parameter data contains the SDU which allows the service user to transmit its PDU to the peer service user entity. For example, the DL-DATA parameter data contains layer 3 (i.e. MiNET) messages.

Note - The operations across the data link layer/layer 3 boundary shall be such that the layer sending a primitive can assume a temporal order of the bits within the parameter data, and that the layer receiving the primitive can reconstruct the information with its assumed temporal order.

2.2 Primitive procedures

For detailed descriptions of the primitive procedures across the layer 3/data link layer boundary, refer to MTS21 [10].

3 Message structures and allocations

3.1 Format convention

3.1.1 Numbering Convention

The basic convention used in this MTS is illustrated in Figure 1/MTS22-C. The bits are grouped into octets. The bits of an octet are shown horizontally and are numbered from 1 to 8. Multiple octets are shown vertically and are numbered from 1 to
n.

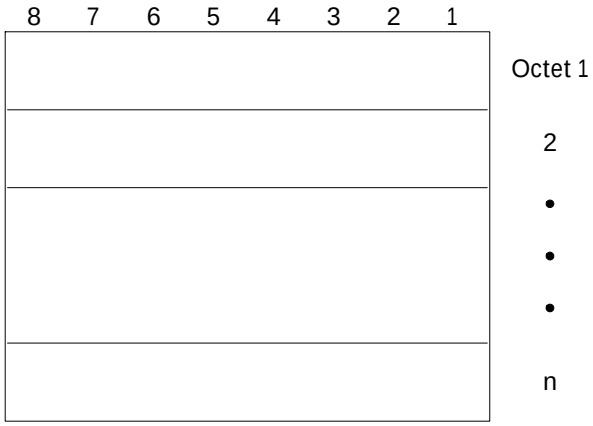


Figure 1/MTS22-C
Format convention

3.1.2 Order of transmission

The octets are transmitted in ascending numerical order. Inside an octet, bit 1 is the first bit to be transmitted.

3.1.3 Field mapping convention

When a field is contained within a single octet, the lowest bit number of the field represents the lowest order value.

When a field spans more than one octet, the order of bit values within each octet progressively decreases as the octet number increases. The lowest bit number associated with the field represents the lowest order value.

For example, a bit number can be identified as a couple (o, b) where o is the octet number and b is the relative bit number within the octet. Figure 2/MTS22-C illustrates a field that spans from bit (1,3) to bit (2,7). The high order bit of the field is mapped on bit (1,3) and the low order bit is mapped on bit (2,7).

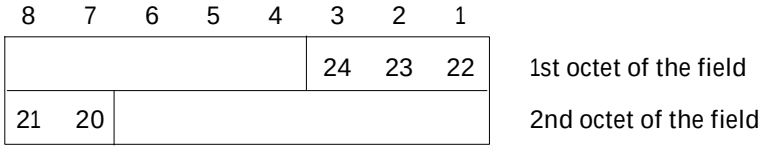


Figure 2/MTS22-C
Field mapping convention

3.2 General format

All MiNET messages shall consist of one or more complete octets, and shall follow the general format shown
below:

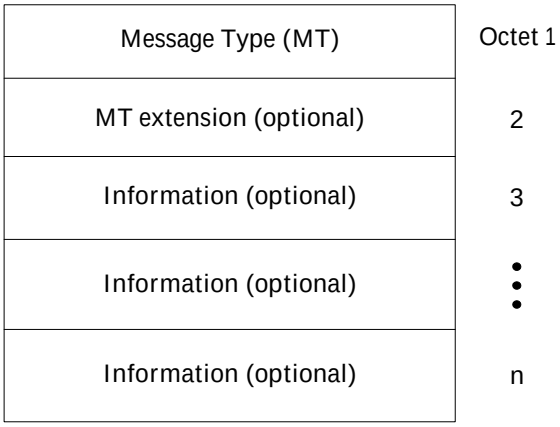


Figure 3/MTS22-C
MiNET message general format

Several values of the initial message type (MT) octet are reserved as extension codes (see § 3.3). The presence of these codes indicates that the MT field extends into the next octet (octet 2).
Variable length messages are assembled as required. The maximum message length is limited by the data link layer to 260 octets, however specific terminals may only allow significantly shorter messages due to buffering constraints.

3.3 MT code allocations

Table 2/MTS22-C shows the MT code allocations based on product types. Values are shown in hexadecimal.

Table 2/MTS22-C
MT code allocations

	Value
Voice sets and consoles (127 values)	00 … 7E
Extension	7F
Data sets (63 values)	80 … BE
Extension	BF
Common (63 values)	C0 … FE
Extension	FF

MT codes and messages within the common block of allocations have applicability to both voice sets and data sets.

Note - Common block codes F3 and F7 are reserved for the Mitel simplified protocol (sP), and shall not be used in the system-to-terminal direction for MiNET. The sP protocol is not described in this specification.

The highest octet value in each block is reserved for an extension code. This allows extension to an additional 256 codes within each block.

3.4 Unsupported messages and message contents

Not all MT codes within each allocation block are defined. Additionally, not all terminals support all MT codes. Unsupported messages received by a terminal should be rejected via a Network Level Message Reject response (see § 4.2.2.2).

If message contains unsupported parameters, the terminal should reject the message via a Network Level Message Reject response (see § 4.2.2.2).

3.5 Terminology

In general, for the purpose of this MTS, the term command shall be used for messages from the system to the terminal, whereas the term response shall be used for messages from the terminal to the system. Examples of exceptions to this convention are the datagram message and data transport messages, which are neither commands nor responses.

4 Messages common to all terminals

In the message descriptions following, the MT (Message Type) code values are presented as hexadecimal values, and are shown in brackets following the message title.

4.1 Allocated messages

Table 3/MTS22-C lists all messages defined within the common allocation block.

Table 3/MTS22-C
Common block message allocation

MT	Message
C0	Request Set ID
C1	Report Set ID
C2	Network Level Message Reject
C3	Read Set Memory
C4	Report Set Memory
C5	Request Maintenance Data
C6	Send Maintenance Data
C7	Request Set Reset
C8	Load Address
C9	Load Data
CA	Variable Length Load Data
CB	Checksum Request
CC	Checksum Reply
CD	Run
CE	Write Set Memory
CF	Define Channel and MiLINK Parameters
D0	Configure Hardware
D1	Report Channel and MiLINK Parameters
D2	Datagram
D3	Local Programmable Key Control
D4	Miscellaneous Indicator Control
D5	Set Relay
D6	Pager Module Ringing
D7	Report Config Change
D8	Send Set Serial Number
D9	PIN Reply
DA	PIN Request
DB	Set Lockup Request
DC to FE	Reserved
FF	Extension code

4.2 Message descriptions

4.2.1 Commands from system to terminal

4.2.11 Request Set ID (C0)

The format of this command shall be as shown in Figure 4/MTS22-C.
This command shall be used by the system to determine the identity of attached terminals. The terminal shall reply with a Report Set ID response (see § 4.2.2.1).

Note - This command is generally sent as the result of a system reset.

All terminals shall support this command.

Figure 4/MTS22-C
Request Set ID command format

4.2.1.2 Read Set Memory (C3)

The format of this command shall be as shown in Figure 5/MTS22-C.
This command shall be used by the system to read specific memory locations or hardware values within terminals.

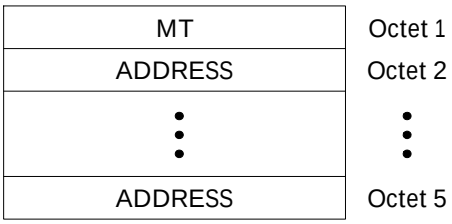


Figure 5/MTS22-C
Read Set Memory command format

Octets 2 through 5 of the command specify a 32-bit start address. Octet 2 shall represent the highest order address octet.

Note - The number of consecutive octets returned to the system as a result of this command is terminal-dependent.

4.2.1.3 Request Maintenance Data (C5)

The format of this command shall be as shown in Figure 6/MTS22-C.
This command shall be used by the system to request information from a terminal for maintenance purposes. When this command is supported in a terminal, a Send Maintenance Data response shall always be sent in reply (see § 4.2.2.4).

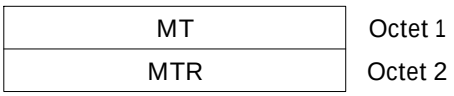


Figure 6/MTS22-C
Request Maintenance Data command format

The MTR (Maintenance Request) octet specifies the maintenance request to be performed by the terminal. Table 4/MTS22-C lists the codings, meanings and expected responses for all defined MTR values. MTR octet values are shown in hexadecimal.

Table 4/MTS22-C
MTR octet codings and responses

MTR	Request to terminal	Response from terminal
00	End test flag	None – any loopbacks disconnected
01	EPROM checksum request	EPROM checksum pass/fail
02	PCM loopback request	PCM loopback established

03	Voice analog loopback request	Analog loopback established
04	Dataset data loopback	Dataset loopback established
05	Local DTE loopback	Local DTE loopback signal asserted
06	Bphone test request	Bphone test pass/fail
07	Hardware revision request	Return one-octet revision
08	Software revision request	Return one-octet revision
09	RAM-based checksum prompt status	Not initiated by system
0A	Remote DTE loopback	Remote DTE loopback signal asserted
0B	Transducer register contents	Return register contents
0C	RAM-based character checksum status	Not initiated by system
0D	Boot firmware revision request	BOOT firmware revision level: format and response same as Send Maintenance Data (0xC6), MTR-08

The dataset data loopback is formed between the dataset's output to the attached DTE and the input from the attached DTE (i.e. the dataset output stream replaces the input stream from the DTE).
The local DTE loopback is accomplished by the dataset signalling a DTE loopback request via an appropriate interface control lead. If supported in the DTE, the data stream from the dataset to the DTE will be reflected back to the dataset via an internal loopback in the DTE.
Note - In some cases, the system may also need to wait for an appropriate indication from the DTE prior to commencing transmission of test data.
The suite of tests available to the system is dependent on the nature of the terminal. For example, data loopback tests have no meaning for a telephone set.

4.2.1.4 Request Set Reset (C7)

The format of this command shall be as shown in Figure 7/MTS22-C.
This command shall be used by the system to direct a reset of a terminal. The terminal's behaviour shall be equivalent to power-up.

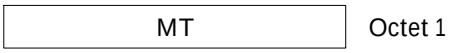


Figure 7/MTS22-C
Request Set Reset command format

Note - When directed to a telephone set, this command and the resulting behaviour of the set shall not affect any peripherals which may be attached to the set's MiLINK port.
A telephone set will take on the following default configuration following power up, reset, loss of DNIC synchronization, or receipt of a Request Set Reset command:

- DTMF tones OFF for consoles, ON for others;
- all output transducers OFF;
- all delay updates cleared;
- buffer number 1 mapped to the screen;
- cursor locations set at (1,1);
- cursors OFF; and
- B1 channel to be used.

The following set functions will be in an unknown state (i.e. may be temporarily used by the set for a special purpose, such as an indication of power up state):

- line LCDs;
- LEDs;

- LCD display;
- buffer contents; and
- volumes and ringer combination.

Note - A possible exception to the above is handset volume, which may be retained by the set during a directed reset.

4.2.1.5 Write Set Memory (CE)

The format of this command shall be as shown in Figure 8/MTS22-C.
This command shall be used by the system to modify memory within a terminal.

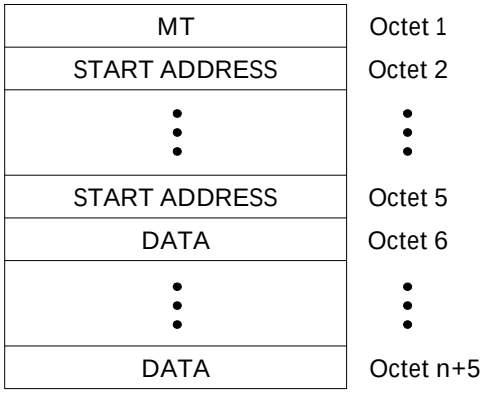


Figure 8/MTS22-C

Write Set Memory message format

Octets 2 through 5 specify a 32-bit start address. Octet 2 shall represent the highest order address octet. Octets 6 through n+5 contain variable length load data. The number of octets of data which can be written via this command is terminal dependent. For datasets, the value of n shall be 9.

4.2.1.6 Define Channel And MiLINK Parameters (CF)

The format of this command shall be as shown in Figure 9/MTS22-C.
This command shall be used by the system to perform one or more of the following functions on a terminal:

a) control MiLINK locking;
b) specify the B channel to be used; and/or
c) specify the data link layer address (see also MTS21 [10]).

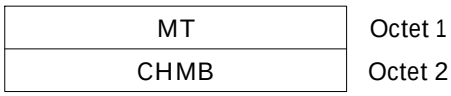


Figure 9/MTS22-C

Define Channel And MiLINK Parameters command format

The CHMB (Channel and MiLINK octet) octet is defined as follows:

b8 = MiLINK lock control

0 no change
1 unlock MiLINK

b7 b6= B channel assignment

0 0 no change
0 1 use B1 for primary
1 0 use B2 for primary
1 1 release B1 and B2 channels

b5	b4	b3	b2	b1	= MiLINK data link layer address allocation
0	0	0	0	0	no change
0	0	0	0	1	reserved for data sets
0	0	0	1	0	available
		to			
1	1	1	1	0	available
1	1	1	1	1	reserved for address negotiation (see MTS21 [10])

Note- See § 3.3/MTS21 for the format of address allocation.

4.2.1.7Configure Hardware (D0)

The format of this command shall be as shown in Figure 10/MTS22-C.

This command shall be used by the system to initialize and/or control any allowable hardware parameter or feature within a
peripheral.

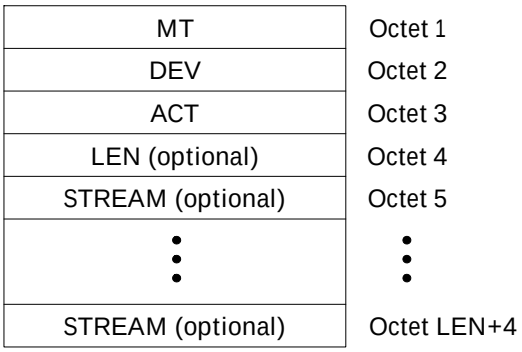


Figure 10/MTS22-C

Configure Hardware command format

The LEN (Length) octet, when used, shall specify the length of the STREAM field in octets. Valid values for LEN shall fall within the range 0 to 255 (decimal). The LEN field shall not be used when the value of ACT is 5 (disable) or 6 (return to default).

The STREAM field, when used, is dependent on the value of the DEV and ACT octets. The STREAM field shall not be used when the value of ACT is 5 or 6.

The DEV (Device) octet specifies the device type to be acted upon, and is defined as follows:

00 = hard reset of phone (if act == 6)
01 = volume keys
02 = CODEC law and format control
03 = flashhook timing
04 = unused
05 = ringing frequency and level
06 = filter coefficients
07 = HC11 flash (memory device)
08 = Mantis flash (memory device)

09 = Headset override default rules (no LEN or STREAM bytes accepted)[1]

The ACT (Action) octet specifies the type of action to be performed, and is defined as follows:

1 = read
2 = write
3 = execute
4 = enable
5 = disable
6 = return to default
7 = fill buffer with data to be programmed into set's flash
    LEN - length of data
    STREAM - first byte: where in buffer to write the data, other bytes: the data
8 = program buffer
    LEN - size of pointer to follow (16-bit or 32-bit)
    STREAM – flash address to write buffer (16-bit: offset, 32-bit: address)

The following sections provide some specific examples of the use of this command.

[1] Normally when a headse is plugged into a telephone, the audio path is directed to the headset, and there is no way to use the handset while the headset is plugged in. This command overrides the normal behavior of the phone, and if this override is enabled, the audio path will be directed to the handset. If the override is disabled, the audio path will revert to the headset.

4.2.1.7.1 Disabling or reassignment of volume keys

For special applications, it may be desirable to de-assign the volume control function assigned to the volume or arrow keys, allowing those keys to be used in a manner similar to other non-DTMF keys. This is accomplished via the configure hardware command using the following codings:

DEV = 01 (volume keys)
ACT = 5 (disable)

Note - For the above value of ACT, the LEN and STREAM fields shall not be used.

It is also possible to reassign the volume function to a different set of keys. This is accomplished via the configure hardware command using the following codings:

DEV = 01 (volume keys)
ACT = 3 (execute)
LEN = 2 (2 octets to follow)
STREAM = volume-up key, volume-down key (new keys assigned to volume)

Note - The values for individual keys are terminal dependent.

The following coding for the configure hardware command returns the volume functions to the default keys:

DEV = 01 (volume keys)
ACT = 6 (return to default)

Note - For the above value of ACT, the LEN and STREAM fields shall not be used.

4.2.1.7.2 Setting CODEC law and format

The terminal's CODEC companding law and PCM format are set via the configure hardware command using the following codings:

DEV = 02 (CODEC law and format control)
ACT = 3 (execute)
LEN = 1 (1 octet to follow)
STREAM = law and format control field

The STREAM octet in this case is defined as follows:

b8	b7	b6	b5	b4	b3	b2	b1
x	x	x	x	x	x	LAW	FMT

where x = don't-care, LAW = companding law, and FMT = PCM format.

The following codings for LAW and FMT shall be used.

LAW	FMT	Meaning
0	0	μ-law with Mitel format (sign + magnitude)
0	1	μ-law with industry standard format (AT&T D3/D4)
1	0	A-law with Mitel format
1	1	A-law with CCITT standard format (G.711/G.712)

4.2.1.7.3 Setting of flash hook timing

It is possible to set the minimum and maximum flashhook time interval via the configure hardware command using the following codings:

DEV = 03 (flashhook timing)
ACT = 3 (execute)
LEN = 2 (2 octets to follow)
STREAM = minimum value, maximum value, both in 10 ms increments

It is possible to enable, disable or set to the default value flashhook detection via the configure hardware command using the following codings:

DEV = 3 (flashhook timing)
ACT = 4 (enable), 5 (disable) or 6 (return to default)

4.2.1.7.4 Setting of ringing frequency and level

It is possible to set the frequency and the level of the ringing via the configure hardware command using the following codings:

DEV = 05 (ringing frequency and level)
ACT = 3 (execute)

LEN = 1 (1 octets to follow)
STREAM = RGENF and RGENV
The STREAM octet in this case is defined as follows:
RGENF RGENV

b8	b7	b6	b5		b4	b3	b2	b1	
0	0	0	0	20 Hz	0	0	0	0	65 V rms
0	0	0	1	25 Hz	0	0	0	1	80 V rms

All other values are reserved for further enhancements.

4.2.1.7.5 Setting of filter coefficients

It is possible to set the filter coefficients via the configure hardware command using the following codings:
DEV = 06 (filter coefficients)
ACT = 3 (execute)
LEN = variable, depending on the type of filter coefficient (range 0 to FF)
STREAM = coefficient values
The coding of the coefficient values in the STREAM octets is product-dependant.

4.2.1.18 Datagram (D2)

The format of this message shall be as shown in Figure 11/MTS22-C.
This message allows the exchange of application-specific information between communicating entities. It may be used for host-to-user (either direction) or user-to-user applications.

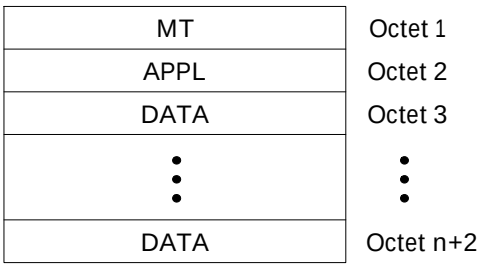


Figure 11/MTS22-C

Datagram message format

The APPL (application code) octet is currently defined as follows (decimal):

0 = not used
1 = host parameters
2 = Analog Line Monitor
3 = Q2000 message encapsulation
4 to 255 = not defined

The length and content of the data field are application dependent. Examples of some possible data field contents are:
- source equipment ID;
- request indication;
- reply indication; and
- field length.

4.2.1.9 Set Relay (D5)

The format of this command shall be as shown in Figure 12/MTS22-C.

This command shall be used by the system to activate/deactivate the indicated relay(s).

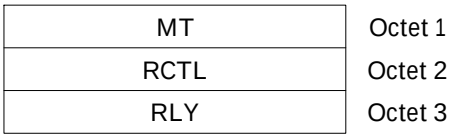


Figure 12/MTS22-C

Set Relay message format

The RCTL (relay control) octet is defined as follows (decimal):

0 = deactivate
1 = activate

The RLY (relay) octet is defined as follows (decimal):

1-254 = relay number
255 = all relays

4.2.1.10 **Pager Module Ringing (D6)**

The format of this command shall be as shown in Figure 13/MTS22-C.

This command shall be used by the system to control a ringing cycle in a device that is controlling a Paging Unit. Cadence pattern and repeat pattern is implementation dependant.

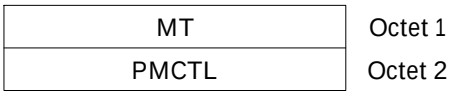


Figure 13/MTS22-C

Pager Module Ringing message format

The PMCTL (pager module control) octet is defined as follows (decimal):

00 = deactivate
01 = activate

4.2.2 Responses from terminal to system

4.2.2.1 Report Set ID (C1)

The format of this response shall be as shown in Figure 14/MTS22-C.

This response shall be sent by a terminal on power-up, and shall also be sent in response to a Request Set ID command (see § 4.2.1.1).

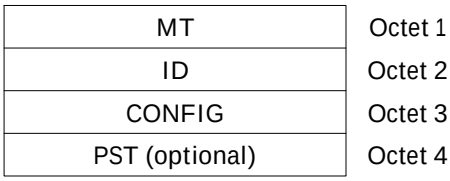


Figure 14/MTS22-C

Report Set ID response format

The values (in hexadecimal) and meanings currently defined for the ID octet are shown below:

Note- Bit 7 of the ID octet is traditionally used to differentiate voice (bit=1) and data (bit=0) terminals.

ID octet:

00 = asynchronous dataset (DS1100)
01 = low speed synchronous dataset (DS2100)
02 = high speed synchronous dataset (DS2200)
03 = IBM 3278 terminal dataset (DS4113)
04 = IBM 3274 controller dataset (DS4122)
05 = Printer port option on DNIC LCD console

06 = MiLINK data module
07 = DNIC universal dataset
08 to 7D = undefined
7E = system node (e.g. line card, communications processor, etc.)
7F = unused
80 = SS3DN
81 = SS4DN
82 = BLF (Busy Lamp Field)
83 = DNIC LCD console
84 = DNIC SS7 (SuperSet 7)
85 = SS401 and SS401+
86 = SS410
87 = SS420
88 = SS430
89 = PC console
8A = AIM (Analogue Interface Module)
8B = MiLINK CAU (Call Announce Unit)
8C = SS 1DN
8D = ALM (Analogue Line Monitor)
8E = MiLINK PKM (Programmable Key Module)
8F = SS 415
90 = DMP (DNIC Music on hold/Pager Module)
91 = Radio Port
92 = SS601+ (600 series SS401+)
93 = SS610 (600 series SS410)
94 = SS620 (600 series SS420)
95 = SS630 (600 series SS430)
96 = SS622 (600 series MiLINK PKM)
97 = SS4001
98 = SS4015
99 = SS4025
9A = SS4125
9B = SS4150
9C = SS4090
9D = PKM12/PKM48
9E = ALI
9F = 4015IP and SS5010
A0 = 4025IP and SS5020
A1 = SS5001
A2 = SS5005
A3 = IP Console
A4 = SS5140 (Webset)
A7 = 4015PC (Soft phone)
A8 = 4025PC (Soft phone)
A9 = 4025VM (Voice mail virtual extension)
AA to FF = reserved

The CONFIG octet shall have the following format:

b8	b7	b6 to b4	b3 to b1
LRB	0	Report ID Reason code	HW Variant

Any terminal supporting the download procedures (see § 4.2.3.1) shall set the LRB (Load Request Bit) bit to 1 when first sending the Report Set ID response after a power-up or reset. Terminals that do not support the download procedures shall always set the LRB bit to 0. The LRB bit shall be set to 0 after the terminal receives a Run command. Any subsequent responses to the Request Set ID command shall have the LRB bit set to 0.

The Report ID Reason code shall be used to indicate the reason for transmission of the Report Set ID response. Terminals that do not support this field shall set all the bits (b6 to b4) to 0. The following reason codes are defined:

Report ID Reason code:

b6	b5	b4	Meaning
0	0	0	Undefined
0	0	1	Hardware reset or power-up
0	1	0	Loss of DNIC synchronization
0	1	1	MiNET initiated reset
1	0	0	ID requested
1	0	1	Software TRAP
1	1	0	Received MiNET reject on D channel
1	1	1	Device specific reset reason

The HW Variant field shall be used to indicate specific variants of a terminal. For example, the asynchronous dataset has the following variants:

1) cartridge;
2) single stand-alone;
3) single rack-mount;
4) dual stand-alone master;
5) dual stand-alone slave;
6) dual rack-mount master; and
7) dual rack-mount slave.

The HW Variant field is therefore specific to each terminal type.
The PST (Power-up Self Test) octet shall be used to report the terminal's hardware device number and the result of power-up self tests as follows:
Bits 8 to 5 = Hardware device number
Bits 4 to 1 = Self test result
For the self test result, the value 0 shall be used to indicate all tests passed. Remaining values from 1 to 15 (decimal) are specific to given terminals.

4.2.2.2 Network Level Message Reject (C2)

The format of this response shall be as shown in Figure 15/MTS22-C.
This response shall be used by a terminal to indicate to the system that a received network message was not within the terminal's repertoire. Any further action shall be initiated by the system.
A data terminal connected on a B channel, upon transmission of this response shall disconnect from the B channel and revert to the D channel (see also § 6.2.2.2.4).

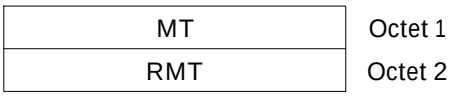


Figure 15/MTS22-C
Network Level Message Reject response
The RMT (Rejected Message Type) octet shall contain the MT code of the unknown message.

4.2.2.3 Report Set Memory (C4)

The format of this response shall be as shown in Figure 16/MTS22-C.
This response shall be sent by a terminal in reply to a Read Set Memory command (see § 4.2.1.2). The data field shall contain n octets beginning at the address specified in the Read Set Memory command. The actual value of n is terminal

dependent.

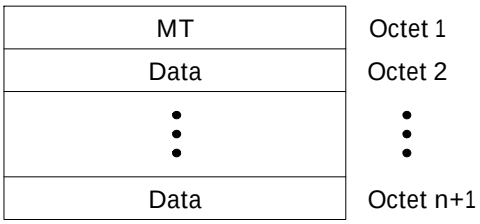


Figure 16/MTS22-C
Report Set Memory Response

4.2.2.4 Send Maintenance Data (C6)

The format of this response shall be as shown in Figure 17/MTS22-C.

This response shall be used by the terminal to reply to a Request Maintenance Data command (see § 4.2.1.3). The terminal may also send the response independently in the event the its firmware detects a failure.

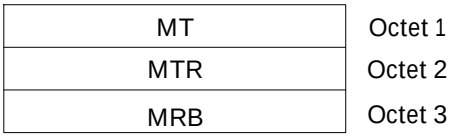


Figure 17/MTS22-C
Send Maintenance Data response format

The MTR (Maintenance Request) octet is defined as follows (hexadecimal):

00 = End test flag
01 = EPROM checksum request
02 = PCM loopback request
03 = Voice analog loopback request
04 = Dataset data loopback
05 = Local DTE loopback
06 = Bphone test request
07 = Hardware stream and revision request
08 = Software stream and revision request (MAIN firmware)
09 = RAM-based checksum prompt status
0A = Remote DTE loopback
0B = Transducer register contents
0C = RAM-based character checksum status

The MRB (Maintenance Response Byte) octet is defined as follows (hexadecimal):

00 = Test (as indicated by MTR) initiated
01 = Test (as indicated by MTR) passed
02 = Test (as indicated by MTR) failed
FF = Test (as indicated by MTR) not supported
If MTR = 7, then MRB shall contain the hardware stream and revision number
If MTR = 8, then MRB shall contain the software stream and revision number

Major Rev				Minor Rev					
D7	D6	D5		D4	D3	D2	D1	D0	
0	0	0	Rev 4.xx	0	0	0	0	0	must never use
0	0	1	Rev 5.xx	0	0	0	0	1	Rev x.00
0	1	0	Rev 6.xx	0	0	0	1	0	Rev x.01
…				…					
1	1	0	Rev 10.xx	1	1	1	1	1	Rev x.30

Table 5/MTS22-C shows the various terminal responses possible for the test cases requested from the system.

Table 5/MTS22-C
Test Request
Responses

System MTR	Terminal MTR	Possible MRB responses
01	01	01, 02, or FF
02	02	00 or FF
03	03	00 or FF
04	04	00, 01, 02, or FF
05	05	00 or FF
06	06	01, 02, or FF
07	07	Hardware stream/revision
08	08	Software stream/revision
Not initiated by system	09	01, 02, or FF
0A	0A	00 or FF
0B	0B	Transducer register contents
Not initiated by system	0C	01, 02, or FF

Test 04, dataset data loopback, may occur in two ways. If the response contains the value 00, then the system shall source the test data and wait for the looped data return in order to determine test pass or failure. If the response contains the values 01 or 02, then the dataset has performed the test locally and has informed the system of the result. No other test action by the system is necessary in this case.
Bits 8-6 of the stream/revision (MTR = 07 or 08) should contain the stream coded as 0 for stream A through 7 for stream H. Bits 5-1 should contain the stream coded in binary format, i.e., 00 for revision 0 through 1F for revision 31).

Note- Existing implementation may encode the stream/revision MRB octet in a different way.

4.2.2.5 Report Channel And MiLINK Parameters (D1)

The format of this response shall be as shown in Figure 18/MTS22-C.
This response shall be used by the terminal to reply to a Define Channel And MiLINK Parameters command (see § 4.2.1.6). It may also be used independently by the terminal to report B channel selection and/or MiLINK status to the
system.

MT	Octet 1
CHMB	Octet 2

Figure 18/MTS22-C
Report Channel And MiLINK Parameters response format

The CHMB (Channel and MiLINK Byte) octet is defined as follows:
b8 = MiLINK information
0 MiLINK normal
1 MiLINK devices locked out
b7 b6 = B channel information
0 0 requested B channel not available

0 1 using B1 for primary
1 0 using B2 for primary
1 1 switched to requested channel

Bits 5 to 1 = Not used

Note - The absence of a response from the terminal shall indicate acceptance of the previous Define Channel And MiLINK Parameters command.

4.2.2.6 Report Config Change (D7)

The format of this message shall be as shown in Figure 19/MTS22-C.

This message is issued by the terminal to report several hardware configuration change events which may occur from time to time. Currently defined examples of such events include connection and disconnection of a headset, changes in availability to the PBX of the codec while the PC is using it..

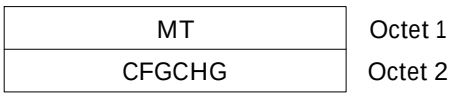


Figure 19/MTS22-C

Report Config Change message format

The CFGCHG octet is defined as follows:

Bits 8 to 4 = reserved

Bit 3 = alternate codec in use by PC

Bit 2 = headset connected

Bit 1 = new configuration indicator

Value 0 = false

Value 1 = true

A configuration change for a component is signaled when the corresponding bit is set to 1. The new state of that component is indicated by the value of bit 1.

4.2.2.7 Send Set Serial Number (D8)

The format of this message shall be as shown in Figure 20/MTS22-C.

If supported by the terminal type, this message shall be used by the terminal to indicate its serial number to the system, and is sent under the same conditions as a Report Set ID (C1).

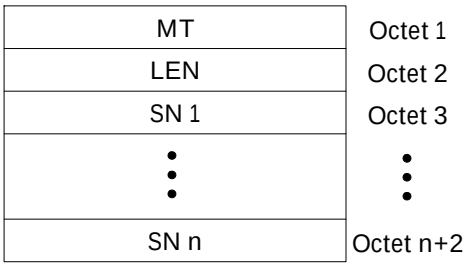


Figure 20/MTS22-C

Send Set Serial Number message format

The following coding shall be used.

LEN = as of the number of SN octets to follow (n)

SN 1-n = Serial Number octet

The LEN (Length) octet shall specify the number of SN octets to follow. The contents of SN octets are determined as follows:

For byte accessible devices, the Most Significant Byte of the serial number shall be transmitted in SN-1, the second Most Significant Byte in SN-2, and so on, with the Least Significant Byte of the serial number transmitted in SN-n.

For devices with a serial interface, the first bit unpacked from the serial interface shall be transmitted in bit 8 of SN-1, the second bit is transmitted in bit 7 of SN-1, the nineth bit in bit 8 of SN 2, and so on. All unused bits of SN-n shall be cleared to 0.
Note - As new serial number devices are used that employ interfaces other than serial, standardization of serial number transmission from these devices shall be added to the definition of this message.

4.2.2.8 PIN Reply (D9)
The format of this message shall be as shown in Figure 21/MTS22-C.
This message will be issued by the terminal in response to a PIN Request (DA).

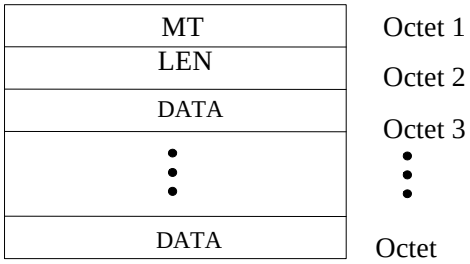


Figure 21/MTS22-C
PIN Reply message format
The following coding shall be used.
LEN = as of the number of data octets to follow (n)
DATA 1-n = DTMF key pressed by user

4.2.2.9 PIN Request (DA)
The format of this message shall be as shown in Figure 22/MTS22-C.
This message may be issued by the system to request a PIN from the phone. The user will then enter digits on the phone keypad, and the phone will respond with a PIN Reply (D9).

Figure 22/MTS22-C

4.2.2.10 Set Lockup Request (DB)
The format of this message shall be as shown in Figure 23/MTS22-C.
This message may be issued by the system to request a set lockup.

Figure 23/MTS22-C

4.2.3 Downloading
4.2.3.1 Download procedures
Immediately after a successful establish request or indication, a terminal shall report its identification via the Report Set ID response (see § 4.2.2.1). If the terminal requires a download, it shall set the Load Request Bit (LRB) to 1. All subsequent Report Set ID responses (e.g. in reply to polls from the system) shall also have this bit set until a Load Address command (see § 4.2.3.2) or Run command (see § 4.2.3.7) is received from the system. After that time, all subsequent Report Set ID responses from the terminal shall have the LRB set to 0.

The system may optionally send a Start B Channel command (see § 6.2.2.1.10) before continuing the download sequence, in order to download the terminal over a B channel.

The download process shall start with a Load Address command from the system. This command specifies the memory address at which the download data storage shall commence. The terminal shall ignore Checksum Request commands (see § 4.2.3.5) and Load Data commands (see § 4.2.3.3) until the first Load Address command is received.

After each valid load message following the first Load Address command, the terminal shall start a timer. The value of this timer shall be 15 seconds. The timer shall be reset on receipt of each subsequent valid load message. The timer shall be reset and stopped on receipt of a Run command. Upon expiry of the timer, the terminal shall reset.

Load Data commands shall contain 27 (decimal) octets of data for DNIC SuperSet 7 consoles, or 13 (decimal) octets of data for datasets. For data fields smaller than the above values, the Variable Length Load Data command (see § 4.2.3.4) shall be employed.

On receipt of a Checksum Request command, the terminal shall respond with a Checksum Reply response (see § 4.2.3.6). The Checksum Request command shall contain the start and end addresses over which the checksum is to be calculated. The command shall also contain the expected checksum value. The Checksum Reply response from the terminal shall contain both the expected and calculated checksum values. If the checksum fails, the system shall reset the terminal using the Request Set Reset command (see § 4.2.1.4). The downloading terminal shall not perform a unilateral reset.

Note - Seemingly erroneous checksums may be normal in multicast downloading.

The checksum shall be determined by summing the words (2-octet values) in memory from the start address to the end address. If the start address is odd, the high-order octet of the word shall be considered equal to 0. Similarly, if the end address is even, the low-order octet of the word shall be considered equal to 0. The resulting checksum value is also a word in length.

A Run command shall be sent at the end of the download data block. The command shall contain the start address for execution. Upon receipt of this command, the terminal shall disconnect from the channel and begin execution at the start address. If the start address field contains the value -1, the terminal shall clear its waiting for load state and continue execution.

For any message received during the above procedures which has an unknown MT code, the terminal shall respond with a Network Level Message Reject response (see § 4.2.2.2).

Figure 21/MTS22-C illustrates a sample download sequence.

Terminals which operate from EPROM shall also send a download request in the Report Set ID response. If the terminal is known by the system to operate from EPROM, the system's reply shall be a Run command which specifies the appropriate start address.

A Checksum Request command will be sent any time a load discontinuity exists, as well as at the end of the load. If the entire load is contiguous, only the final Checksum Request command shall be sent. Load Address commands are not linked to the checksum commands, that is, a Load Address command may be sent at any time.

4.2.3.2 Load Address (C8)

The format of this command shall be as shown in Figure 22/MTS22-C.
Octets 2 through 5 specify a 32-bit start address. Octet 2 shall represent the highest order address octet.
Refer to § 4.2.3.1 for message usage details.

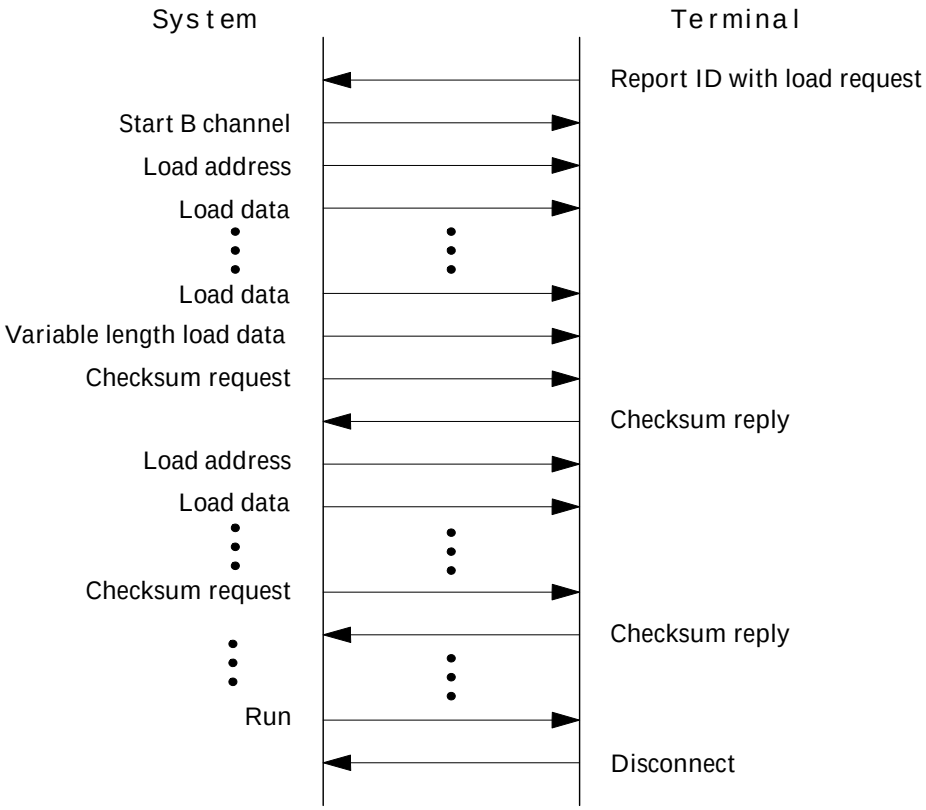


Figure 24/MTS22-C
Sample download sequence

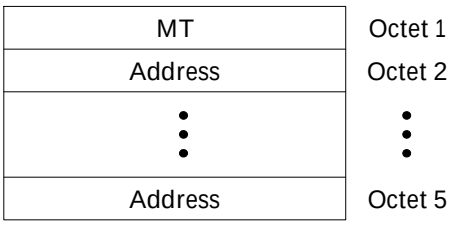


Figure 25/MTS22-C
Load Address command format

4.2.3.3 Load Data (C9)

The format of this command shall be as shown in Figure 26/MTS22-C.
For datasets, the value of n shall be 13 (decimal). For the DNIC SuperSet 7, the value of n shall be 27 (decimal).

Refer to § 4.2.3.1 for command usage details.

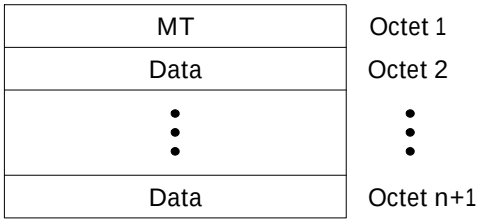


Figure 26/MTS22-C
Load Data command format

4.2.3.4 Variable Length Load Data (CA)

The format of this command shall be as shown in Figure 27/MTS22-C.
Refer to § 4.2.3.1 for command usage details.

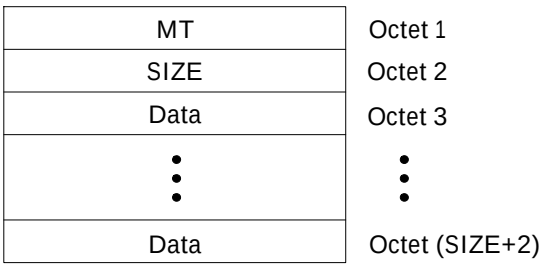


Figure 27/MTS22-C
Variable Length Load Data command format
For datasets, the value of the SIZE octet shall be less than 13 (decimal). For DNIC SuperSet 7, the value of the SIZE octet shall be less than 27 (decimal).

4.2.3.5 Checksum Request (CB)

The format of this command shall be as shown in Figure 25/MTS22-C.
Octets 2 through 5 specify a 32-bit start address. Octet 2 shall represent the highest order address octet.
Octets 6 through 9 specify a 32-bit end address. Octet 6 shall represent the highest order address octet.
Octets 10 and 11 specify a 16-bit expected checksum.
Refer to § 4.2.3.1 for command usage details.

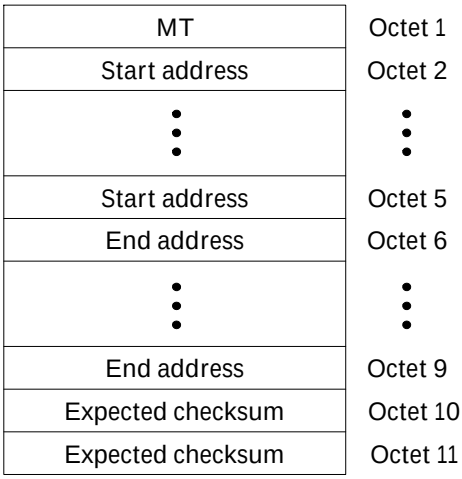


Figure 28/MTS22-C
Checksum Request command format

4.2.3.6 Checksum Reply (CC)

The format of this response shall be as shown in Figure 29/MTS22-C.
Octets 2 through 5 specify a 32-bit start address. Octet 2 shall represent the highest order address octet. Octets 6 through 9 specify a 32-bit end address. Octet 6 shall represent the highest order address octet. Octets 10 and 11 specify the actual 16-bit checksum. Octets 12 and 13 specify the expected 16-bit checksum.
Refer to § 4.2.3.1 for response usage details.

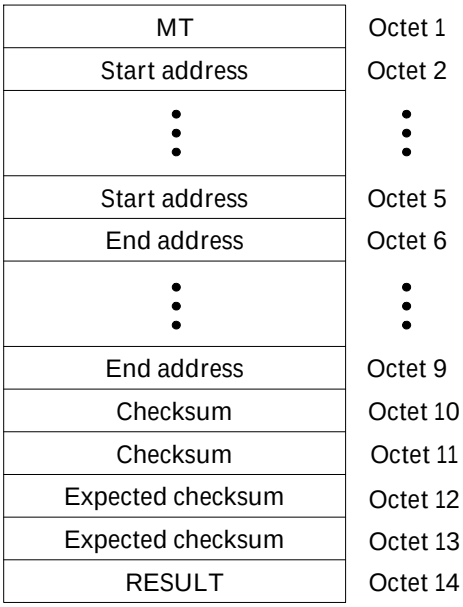


Figure 29/MTS22-C
Checksum Reply response format
The RESULT octet can take on the following values:
1) 0 indicates a bad checksum (actual checksum not equal to expected checksum);
2) 1 indicates a good checksum (actual checksum equal to expected checksum); or

3) 2 indicates an invalid address.

4.2.3.7 Run (CD)

The format of this command shall be as shown in Figure 30/MTS22-C.
Octets 2 through 5 specify a 32-bit start address. Octet 2 shall represent the highest order address octet.
Refer to § 4.2.3.1 for command usage
details.

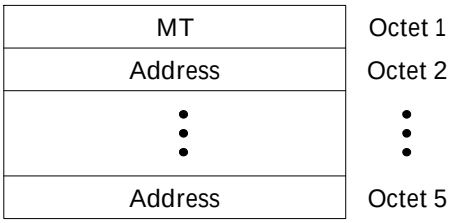


Figure 30/MTS22-C
Run command format

4.2.3.8 Local programmable key control (D3)

It is possible for the host system to control several parameters for a set that is capable of locally storing speed dial numbers and feature codes for its programmable keys (e.g., SS 401+). The format of this command shall be as shown in Figure 31/MTS22-C.

Figure 31/MTS22-C
Local programmable key control command format

MT	Octet 1
LEN	Octet 2
MSK	Octet 3
KCC	Octet 4
EETST	Octet 5
PKC 1	Octet 6
. . .	. . .
PKC n	Octet n + 6

The following codings shall be used:
LEN (Length) = Number of octets to follow (minimum 4)
MSK (Maximum Sequence Keys) = Maximum number of keys to report in a single Send key sequence message (§ 5.2.2.7).
KCC (Key Code Control)

0 - send nothing when key sequence reporting is disabled
1 - send key code for programmable key when key sequence reporting is disabled

EETST (EEPROM Test) = To test the EEPROM. Responds with a link reset if the test passes; no response if failed. The test definition is set-dependent and outside the scope of MTS22.
PKC (Programmable Keys Control) = The LSB of the first octet represents programmable key 1 while the MSB represents key 8. On the second octet, the LSB represents key 9 and so on. Each bit shall be coded as follows:
0 = disable key sequence reporting
1 = enable key sequence reporting

4.2.3.9 Miscellaneous indicator control (D4)

The format of this command shall be as shown in Figure 32/MTS22-C.

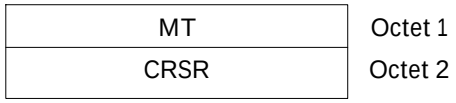


Figure 32/MTS22-C

Miscellaneous indicator control command format

It is possible to control the miscellaneous indicator on a set using the CRSR as defined in § 5.2.1.1.5.

5 Voice sets and consoles messages

In the message descriptions following, the MT (Message Type) codes are presented as hexadecimal values, and are shown in brackets following the messages titles.

5.1 Allocated messages

Table 6/MTS22-C lists all messages defined within the voice sets and consoles allocation block. MT code values are shown in hexadecimal.

Table 6/MTS22-C
Voice sets and consoles message allocation

MT	**Command (system to terminal)**	**Response (terminal to system)**
01	Send String	Send QWERTY Key
02	Send Individual Character	Send Scan Key
03	Send Individual Flashing Character	Acknowledge DTMF
04	Position Cursor	Report Cursor Position
05	Shift Display Field Left	Send Key Sequence
06	Shift Display Field Right	Not used
07	Roll Display Field Left	Store Ring Combination
08	Roll Display Field Right	Store Handsfree Volume
09	Clear Display Line	Store Ringing Volume
0A	Clear Field	Store Music Volume
0B	Flash Field ON	Not used
0C	Flash Field OFF	Not used
0D	Not used	Not used
0E	Display Block Cursor	Not used
0F	Display Underscore Cursor	Not used
10	Blank Cursor	Not used
11	Copy Buffer Line	Not used
12	Clear Screen	Not used
13	Delay Update of a Field	Not used
14	Display Prompt	Not used
15	Store Prompt	Not used
16	Send Prompt	Not used
17	Deactivate Prompt	Not used
18	Display Buffer	Not used
19	Set LCD Indicator	Not used
1A	Load Flash Rate	Not used
1B	Send Signal	Not used
1C	Ringer Cadence High Continuous	Not used
1D	Ringer Cadence High Pulsed	Not used
1E	Ringer Cadence Low Continuous	Not used
1F	Ringer Cadence Low Pulsed	Not used
20	Send DTMF	Not used
21	Not used	Not used
22	Request Cursor Position	Not used
23	Start Precise Ringer Cadencing	Not used
24	Set CODEC Gain	Not used
25	Reserved	Not used
26	Start Voice Set Ringer	Not used
27	Define Initial Conditions	Not used
28	Define Audio Mode	Not used
29	Send Data Module Command	Not used
2A	Delete Character	Not used
2B	Cancel Delay Update Of a Field	Not used

2C	Initialize Display	Not used
2D	Select Stored Prompt	Not used
2E	Define 5 x 7 Special Character	Not used
2F	Not used	Not used
30	Define Character Pattern	Not used
31	Not used	Not used
32	Not used	Not used
33	Setup	Not used
34	Start New Call Tone	Not used
35	Wireless	Not used
36	Map Character To Unicode	Not used
37-4E	Not used	Not used
50	Go To URL	Not used
51	Send Directory Number	Not used
52	Send Label String	Not used
53-7D	Not used	Not used
7E	Keepalive Poll	Link Layer Ack

5.2 Message descriptions

5.2.1 Commands from system to terminal

5.2.1.1 General format and definitions of elements

The general format for messages within the voice sets and consoles allocation block in the direction system to terminal shall be as shown in Figure 33/MTS22-C.

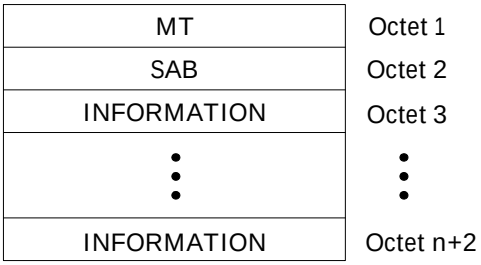


Figure 33/MTS22-C

Voice sets and consoles general command

Note - The value of n is a function of the specific message type. Refer to the detailed descriptions following.

The SAB (Special Attribute Byte) octet can have the following types:

1) BCCB = Buffer / Cursor Control Byte
2) SBDB = Source Buffer / Destination Buffer
3) NPBN = Number of Prompts / Buffer Number
4) PPBN = Prompt Position / Buffer Number
5) CRSR = Circle Rate / Square Rate
6) CSRN = Circle / Square Rate Number
7) HCB = Hardware Control Byte
8) CPB = Cadence Pattern Byte

Note - Not all commands require a SAB octet.

Detailed definitions of the above SAB octet types are given in §§ 5.2.1.1.1 to 5.2.1.1.8.

5.2.1.1.1 Buffer / Cursor Control Byte (BCCB)

The BCCB octet is defined as follows:
b8 = Return Cursor
0 do not return cursor
1 return cursor (leave cursor where it was at start of operation)
b7 = (Y,X) Or Cursor Number
0 use (Y,X)
1 use cursor number
Bits 6, 5, 4 = Cursor Number
Values 0 to 7 = cursor numbers 1 to 8
Bits 3, 2, 1 = Buffer Number
Value 0 = all
Values 1 to 7 = buffers 1 to 7
The Return Cursor bit defines what to do with the cursor position following the execution of a given Minet command. There are 2 possibilities:

a) the cursor specified is to be returned to it's position prior to the execution of the current Minet command;

b) the cursor shall be positioned right of the affected field unless otherwise indicated by the individual Minet command as described in this document.

The (Y,X) Or Cursor Number bit defines whether the operation is to take place at the (Y,X) position given or the current cursor position specified by the cursor number parameter. If the current cursor position is specified, then Y and X parameters shall not be contained in the message. The minimum value of Y and X is 1. The maximum values are a function of the specific terminal type.
Cursor Number specifies the cursor that is to be used for displaying the information. Up to 8 cursors may be supported

Buffer Number specifies the internal RAM buffer to which a string is to apply. If this buffer is currently the one displayed, then the display shall also be updated. Buffer number 0 signifies that all buffers shall be affected by the operation. Up to 7 buffers may be supported.

5.2.1.1.2 Source Buffer / Destination Buffer (SBDB)

The SBDB octet is defined as follows:
Bits 8, 7 = unused
Bits 6, 5, 4 = Source Buffer Number
Value 0 = reserved
Values 1 to 7 = source buffers 1 to 7
Bits 3, 2, 1 = Destination Buffer Number
Value 0 = all destination buffers
Values 1 to 7 = destination buffers 1 to 7

5.2.1.1.3 Number of Prompts / Buffer Number (NPBN)

The NPBN octet is defined as follows
Bits 8 to 4 = Number Of Prompts (0 to 31 decimal)
Bits 3, 2, 1 = Buffer Number
Value 0 = all buffers
Values 1 to 7 = buffer numbers 1 to 7

5.2.1.1.4 Prompt Position / Buffer Number (PPBN)

The PPBN octet is defined as follows:
Bits 8 to 4 = Prompt Position (values are shown in decimal)
Value 0 = reserved
Values 1 to 31 = prompt position number
Bits 3, 2, 1 = Buffer Number
Value 0 = all buffers
Values 1 to 7 = buffer numbers 1 to 7

5.2.1.1.5 Circle Rate / Square Rate (CRSR)

The CRSR octet is defined as follows:
Bits 8 to 5 = Circle Rate
Bits 4 to 1 = Square Rate
The values listed below (decimal) are defined for circle and square rates:
0 = OFF solid
1 = ON solid
2 = rate 2 (250 ms OFF, 250 ms ON)
3 = rate 3 (500 ms OFF, 500 ms ON)
4 = rate 4 (438 ms OFF, 62 ms ON)
5 = rate 5 (62 ms OFF, 438 ms ON)
6 to 13 = pre-definable rates
14 = rate 14 (125 ms OFF, 125 ms ON)
15 = not valid for circle, but for square means square flashes at inverse of circle

5.2.1.1.6 Circle / Square Rate Number (CSRN)

The CSRN octet is defined as follows (values are shown in decimal):
Bits 8 to 5 = unused
Bits 4 to 1 = Rate Number
Values 0 to 5, 14, 15 = reserved
Values 6 to 13 = rates 6 to 13

5.2.1.1.7 Hardware Control Byte (HCB)

Two HCB octets, HCB0 and HCB1, are defined as shown below.
The defined hardware functions, when set to 1 in bits 6 to 1, shall be turned ON or OFF depending on the state of bit 8.
The format of the HCB0 octet is as follows:

b8	b7	b6	b5	b4	b3	b2	b1
ON/OFF	0	HS	HFS	HFMF	DG	NC	RNG

HS = handset microphone (see note 1)
HFS = handsfree speaker
HFMF = handsfree microphone function (see note 2)
DG = DTMF generator (see note 3)
NC = noise cancellation during handsfree operation
RNG = ringer

Note 1- The purpose of the HS bit is to control the handset microphone, earpiece and sidetone together.
Note 2- The purpose of the HFMF bit is to control both the handsfree microphone and the handsfree algorithm.
Note 3- When the DG bit is enabled, the DTMF generator sends tones to all active local transducers and to the line as PCM. Disabling the DTMF generator shall not affect a key depression already in progress (i.e. shall only affect the next key depression).

The format of the HCB1 octet is as follows:

b8	b7	b6	b5	b4	b3	b2	b1
ON/OFF	1	0	HFM	LIND	HSNV	SIDE	EA

HFM = handsfree microphone (see note 1)
LIND = line DTMF (see note 2)
HSNV = handset nominal volume (see note 3)
SIDE = sidetone
EAR = earpiece

Note 1- The purpose of the HFM bit is to control the handsfree microphone only and therefore is different from the HFMF bit in HCB0. The latter bit controls both the handsfree microphone and the handsfree algorithm at the same time.

Note 2- For special applications, the LIND bit may be used to disable DTMF transmission to the line. See also the description above for the HCB0 DG bit.

Note 3- When the HSNV bit and the ON/OFF bit are both set to 1, the handset volume is returned to its nominal value. The HSNV bit has no effect when the ON/OFF bit is set to 0.

5.2.1.1.8 Cadence Pattern Byte (CPB)

Each of the bits in the CPB octet defines a variable ringing time period. The terminal's ringer will be ON during each time period for which the corresponding bit is set to 1, and will be off when the bit is set to 0. The value of the time period is dependant of the message: the default value is 200 ms, unless otherwise stated by a specific message. For example, the time period is defined as 100 ms for Ringer Cadence High Continuous (1C), and is variable for Start Precise Ringer Cadencing (23).

5.2.1.1.9 Extended ASCII character set

Some messages (e.g. send string) employ an extended ASCII character set. The character set fits within a single octet, and is defined as follows (values are shown in hexadecimal):

00 to 7F = standard ASCII character set (as defined in ANSI X3.4 [1])

80-9F Mitel special ASCII characters

- 80 = Solid block
- 81 = Right arrow
- 82 = Left arrow
- 83 = Small block
- 84 = Tilde
- 88 = u grave
- 89 = i circumflex
- 90 = multiply sign
- 91 = decimal point
- 92 = i umlaut
- 93 = u umlaut
- 9A = a grave
- 9B = e acute
- 9C = e umlaut
- 9D = e circumflex
- 9E = u circumflex
- 9F = c cedilla

A0-A8 Downloadable characters

- A0 = a umlaut
- A1 = o umlaut
- A2 = a acute
- A3 = i acute
- A4 = o acute
- A5 = u acute
- A6 = n tilde
- A7 = Inverted exclamation point
- A8 = Inverted question mark

10-17 Characters defined in firmware

- 10 = a umlaut
- 11 = n umlaut
- 12 = o umlaut
- 13 = u umlaut
- 14 = Solid block
- 15 = Divide sign
- 16 = Cent sign
- 17 = Super star

00-07 Downloadable characters

- 00 = c cedilla
- 01 = e grave
- 02 = e acute
- 03 = u circumflex

04 = a grave
05 = a acute
06 = i acute
07 = o acute
C0 = start of a flashing string
C1 = end of a flashing string

5.2.1.2 Send String (01)
This command has two possible formats, as shown in Figure 34/MTS22-C.

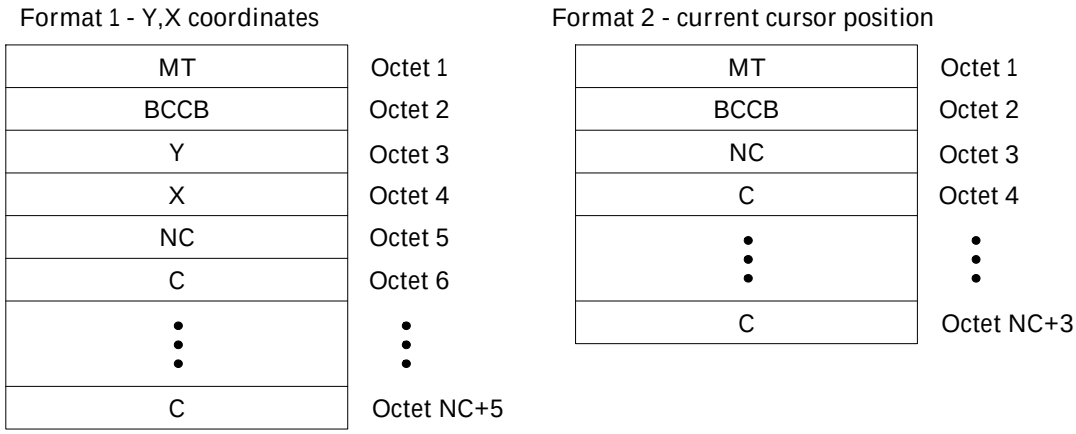


Figure 34/MTS22-C
Send String command format
Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.
The Y octet specifies the vertical line position, and may take on the following values (decimal):
0 = reserved
1 to 255 = valid line numbers
The X octet specifies the horizontal character position, and may take on the following values (decimal):
0 = reserved
1 to 255 = valid character numbers

Note - The top left coordinates of a screen shall be defined as (1,1). The effect of Y,X values which are out of range for a given terminal will be indeterminate.

The NC octet specifies the number of displayable character octets to follow. The NC octet may take on values in the range 0 to 255 (decimal).
The C octets specify the displayable characters, which shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9).

Note - Terminals shall support a maximum string length which is at least 2 characters longer than their screen width. This allows at least one flash sequence to be imbedded in the string.

In format 1, the Y,X parameters specify the starting screen coordinates at which the character string is to be displayed. The Y,X parameters shall be absent from the message (as in format 2) when the (Y,X) or cursor number bit in the BCCB is set to 1.

Note - If the Return Cursor bit in the BCCB octet is set to 1 then, after display of the character string, the cursor shall be returned to its original position. If the Return Cursor bit in the BCCB is set to 0 then, after display of the character string, the cursor shall be left at the end of the displayed string.

5.2.1.3 Send Individual Character (02)

This command has two possible formats, as shown in Figure 35/MTS22-C.
This command shall be used by the system to send an individual character to a particular point on a terminal's
display.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
C	Octet 5

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
C	Octet 3

Figure 35/MTS22-C
Send Individual Character command format

Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.
Use of the Y,X parameters and relationships to the BCCB octet are as defined in § 5.2.1.2.
The C octet specifies the displayable character, which shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9).

5.2.1.4 Send Individual Flashing Character (03)

This command has two possible formats, as shown in Figure 36/MTS22-C.
This command shall be used by the system to send an individual flashing character to a particular point on a terminal's display.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
C	Octet 5

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
C	Octet 3

Figure 36/MTS22-C
Send Individual Flashing Character command format

Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet. Use of the Y,X parameters and relationships to the BCCB are as defined in § 5.2.1.2.
The C octet specifies the displayable character, which shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9).

Note - The flash attribute shall remain with the associated character, regardless of operations such as shift and roll.

5.2.1.5 Position Cursor (04)

The format of this command shall be as shown in Figure 37/MTS22-C.

This command shall be used by the system to position the cursor at a particular point on the terminal's display.

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4

Figure 37/MTS22-C
Position Cursor command format
Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.
Use of the Y,X parameters is as defined in § 5.2.1.2. The Cursor Number bits in the BCCB specify the cursor to be affected.

Note - For this command, the Return Cursor bit in the BCCB octet has no meaning.

5.2.1.6 Shift Display Field Left (05)

This command has two possible formats, as shown in Figure 38/MTS22-C.
This command shall be used by the system to shift a terminal's display one character position to the left and add a new character to the right. The cursor will be returned to the position it occupied before the operation was executed. The field to be shifted must be a subset of one line on the display.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
FL	Octet 5
C	Octet 6

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
FL	Octet 3
C	Octet 4

Figure 38/MTS22-C
Shift Display Field Left command format
The Buffer Number parameter in the BCCB (see § 5.2.1.1.1) specifies the internal buffer to which the operation applies.

Note - If the Buffer Number value indicates all buffers, this command shall be ignored by the terminal.

The Y,X octets specify the leftmost character position of the affected string. The range of values for the Y,X octets is defined in § 5.2.1.2.
The FL octet specifies the field length of the character string to be shifted. The value of the FL octet may occupy the range 0 to 255 (decimal).
The C octet specifies the character to be inserted. The character shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9).

Note - As an example, given the field ABCDEF, the message Shift Display Field Left (+ Z) will result in the display BCDEFZ.

5.2.1.7 Shift Display Field Right (06)

The command formats and codings shall be as defined in § 5.2.1.6.

This command shall be used by the system to shift a terminal's display one character position to the right and add a new character to the left. The cursor will be returned to the position it occupied before the operation was executed. The field to be shifted must be a subset of one line on the display.

Note1 - As an example, given the field ABCDEF, the command Shift Display Field Right (+ Z) will result in the display ZABCDE.
Note2 - If the BCCB octet's Buffer Number value indicates all buffers, this command shall be ignored by the terminal.

5.2.1.8 Roll Display Field Left (07)

This command has two possible formats, as shown in Figure 39/MTS22-C.
This command shall be used by the system to roll a terminal's display one character position to the left. The character at the left end of the selected string prior to the operation is rolled into the right-most position after the operation. No character insertion is supported by this message.The cursor will be returned to the position it occupied before the operation was executed. The field to be rolled must be a subset of one line on the
display.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
FL	Octet 5

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
FL	Octet 3

Figure 39/MTS22-C
Roll Display Field Left command format

The Buffer Number parameter in the BCCB (see § 5.2.1.1.1) specifies the internal buffer to which the operation applies.

Note - If the Buffer Number value indicates all buffers, this command shall be ignored by the terminal.

The Y,X coordinates specify the leftmost character position of the affected string. The range of values for the Y,X parameters is defined in § 5.2.1.2.
The FL octet specifies the field length of the character string to be shifted. The value of the FL octet may occupy the range 0 to 255 (decimal).

Note1 - As an example, given the field ABCDEF, the command Roll Display Field Left will result in the display BCDEFA.
Note2 - If the BCCB octet's Buffer Number value indicates all buffers, this command shall be ignored by the terminal.

5.2.1.9 Roll Display Field Right (08)

The command formats and codings shall be as defined in § 5.2.1.8.
This command shall be used by the system to roll a terminal's display one character position to the right. The character at the right end of the selected string prior to the operation is rolled into the left-most position after the operation. No character insertion is supported by this message.The cursor will be returned to the position it occupied before the operation was executed. The field to be rolled must be a subset of one line on the display.

Note - As an example, given the string ABCDEF, the command Roll Display Field Right will result in the display FABCDE.

5.2.1.10 Clear Display Line (09)

The format of this command shall be as shown in Figure 40/MTS22-C.

This command shall be used by the system to clear the specified lines on a terminal's display. If the Return Cursor bit in the BCCB is set to 1, the cursor shall remain in the position it occupied prior to the operation. If the Return Cursor bit is set to 0, the specified cursor shall be displayed at the left-most position of the last line cleared by the
operation.

MT	Octet 1
BCCB	Octet 2
NFL	Octet 3
NLINES	Octet 4

Figure 40/MTS22-C

Clear Display Lines command format

The NFL octet specifies the number of the first display line to be cleared, and may take on the following values (decimal):

0 = reserved
1 to 255 = valid first line numbers

The NLINES octet specifies the total number of display lines to be cleared including the display line specified by the NFL octet. The NLINES octet may take on the following values (decimal):

0 = reserved
1 to 255 = valid number of lines

Note - When this command is used on display lines where prompts are tracked by the terminal, the prompts shall be deactivated.

5.2.1.11 Clear Field (0A)

This command has two possible formats, as shown in Figure 38/MTS22-C.

This command shall be used by the system to clear a specified field on a terminal's display.

Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.

The Y octet specifies the vertical line position, and may take on the following values (decimal):

0 = reserved
1 to 255 = valid line numbers

The X octet specifies the horizontal character position, and may take on the following values (decimal):

0 = reserved
1 to 255 = valid character numbers

If the Return Cursor bit in the BCCB is set to 1, the cursor shall remain in the position it occupied prior to the operation. If the Return Cursor bit is set to 0, the specified cursor shall be displayed in the left most position of the affected field.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
FL	Octet 5

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
FL	Octet 3

Figure 41/MTS22-C

Clear Field command format

The FL (field length) octet specifies the length of the field to be cleared in characters, and can take on values in the range 0 to 255 (decimal).

5.2.1.12 Flash Field ON (0B)

This command has two possible formats, as shown in Figure 42/MTS22-C.

This command shall be used by the system to enable flashing of a specified field on a terminal's display. The flash rate is fixed at 500 ms ON and 500 ms OFF.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
FL	Octet 5

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
FL	Octet 3

Figure 42/MTS22-C

Flash Field ON command format

The BCCB Buffer Number field shall be used to specify the desired buffer. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.

The Y octet specifies the vertical line position, and may take on the following values (decimal):

0 = reserved

1 to 255 = valid line numbers

The X octet specifies the horizontal character position, and may take on the following values (decimal):

0 = reserved

1 to 255 = valid character numbers

The FL (field length) octet specifies the length of the field affected in characters, and can take on values in the range 0 to 255 (decimal).

The cursor shall remain in the position it occupied prior to this operation. The Return Cursor bit in the BCCB shall not affect this operation.

Note- The flash attribute shall remain with the associated character, regardless of operations such as shift and roll.

5.2.1.13 Flash Field OFF (0C)

The command formats and codings shall be as defined in § 5.2.1.12.

This command shall be used by the system to disable flashing of a specified field on a terminal's display. The cursor shall remain in the position it occupied prior to this operation. The Return Cursor bit in the BCCB shall not affect this operation.

5.2.1.14 Display Block Cursor (0E)
The format for this command shall be as shown in Figure 43/MTS22-C.
This command shall be used by the system to cause a block cursor to be displayed at the terminal's current cursor
position.

MT	Octet 1
BCCB	Octet 2

Figure 43/MTS22-C
Display Block Cursor command format
The BCCB octet's Buffer Number and Cursor Number fields shall be used to specify the desired buffer and cursor. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet. This command does not support BCCB use of (Y, X). The terminal should respond with a Network Level Message Reject (C2) if selected.

5.2.1.15 Display Underscore Cursor (0F)
The format and coding for this command shall be as defined in § 5.2.1.14.
This command shall be used by the system to cause an underscore cursor to be displayed at the terminal's current cursor position. This command does not support BCCB use of (Y, X). The terminal should respond with a Network Level Message Reject (C2) if selected.

5.2.1.16 Blank Cursor (10)
The format and coding of this message shall be as defined in § 5.2.1.14.
This command shall be used by the system to blank a cursor currently displayed on a terminal. This command does not support BCCB use of (Y, X). The terminal should respond with a Network Level Message Reject (C2) if selected.

5.2.1.17 Copy Buffer Line (11)
The format of this command shall be as shown in Figure 44/MTS22-C.
This command shall be used by the system to copy a line from one display buffer to another display buffer.

MT	Octet 1
SBDB	Octet 2
YSRC	Octet 3
YDEST	Octet 4

Figure 44/MTS22-C
Copy Buffer Line command format
The SBDB octet specifies the Source Buffer Number and Destination Buffer Number (see § 5.2.1.1.2).
The YSRC octet defines the source buffer's vertical line number, and may take on values (decimal) as follows:

0 = reserved
1 to 255 = valid line numbers

The YDEST octet defines the destination buffer's vertical line number, and may take on values (decimal) as follows:

0 = reserved
1 to 255 = valid line numbers

The cursor shall remain in the position it occupied prior to this operation.

5.2.1.18 Clear Screen (12)
The format of this command shall be as shown in Figure 45/MTS22-C.
This command is used by the system to clear a terminal's screen buffer. The specified cursor shall be placed at the top left corner of the display after the operation.

The BCCB octet's Buffer Number and Cursor Number fields define the desired buffer and cursor. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.
Note- When a terminal supports prompts, this command shall also deactivate all prompts.

MT	Octet 1
BCCB	Octet 2

Figure 45/MTS22-C
Clear Screen command format

5.2.1.19 Delay Update Of A Field (13)
This command has two possible formats, as shown in Figure 46/MTS22-C.
This command shall be used by the system to update a field on a terminal's display after a specified time delay.

Note - This command could be used, for example, to allow a user to read new displayed data before replacing the field with another.

Format 1 - Y,X coordinates

MT	Octet 1
BCCB	Octet 2
Y	Octet 3
X	Octet 4
FL	Octet 5
T	Octet 6
NC	Octet 7
C	Octet 8
⋮	
C	Octet n+7

Format 2 - current cursor position

MT	Octet 1
BCCB	Octet 2
FL	Octet 3
T	Octet 4
NC	Octet 5
C	Octet 6
⋮	
C	Octet n+5

Figure 46/MTS22-C
Delay Update Of A Field command format
The BCCB octet's Buffer Number specifies the desired buffer. The Buffer Number value shall not be equal to 0. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet. If Do Not Return Cursor is selected in the BCCB the cursor shall be positioned immediately right of the update.

The Y octet specifies the vertical line position, and may take on the values (decimal):
0 = reserved
1 to 255 = valid line numbers

The X octet specifies the horizontal character position, and may take on the values (decimal):
0 = reserved
1 to 255 = valid character numbers

The FL (Field Length) octet specifies, in characters, the length of the field to be updated. The FL octet may take on values in the range 0 to 255 (decimal).
The T (Time) octet specifies, in units of 50 ms, the time delay to be used prior to update of the field. The T octet may take on values in the range 0 to 255 (decimal).

The NC (Number of Characters) octet specifies the number of displayable characters to follow. The NC octet may take on values in the range 0 to 255 (decimal).

The C octet specifies the character to be displayed. The character shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9). If the number of characters is less than the field length specified by the FL octet, then the remainder of the field shall be filled with blanks.

5.2.1.20 Display Prompt (14)

The format of this command shall be as shown in Figure 47/MTS22-C.

This command shall be used by the system to light soft keys with text stored previously in the terminal. If the soft key is being lit on a line that had previously displayed text, then all the boxes for any activated soft keys on that line shall be re-displayed.

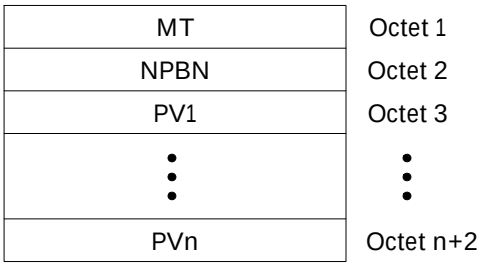


Figure 47/MTS22-C

Display Prompts command format

The NPBN octet's Number of Prompts and Buffer Number fields specify the number of prompts and buffer number affected by the command. The number of prompts is limited by the NPBN octet to 31 (decimal) per command. Refer to § 5.2.1.1.3 for a detailed description of the NPBN octet.

The PV (Prompt Value) octets (1 to n) specify the individual prompt values. The prompts shall be listed in the order of actual display position on the terminal. The PV octet may take on the following values (decimal):

0 to 252 = valid prompt values
253 = deactivate prompt
254 = blank prompt position
255 = no change to prompt position

Note - For PV value 254, only the prompt text shall be blanked. The associated delimiters and boxes shall remain unaffected.

5.2.1.21 Store Prompt (15)

The format of this command shall be as shown in Figure 48/MTS22-C.

This command shall be used by the system to store prompt text in a terminal's internal table. The specified Prompt Value acts as an index into the table.

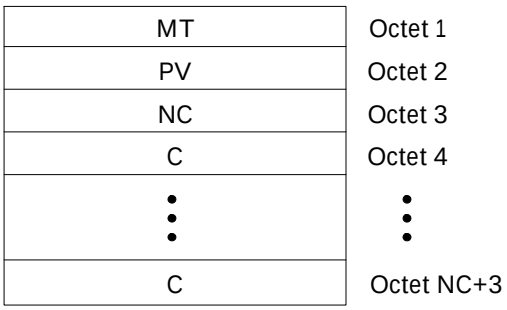


Figure 48/MTS22-C
Store Prompt command format

Note - This command does not employ a SAB octet.

The PV (Prompt Value) octet specifies the individual prompt value, and may take on the following values (decimal):

0 to 254 = valid prompt values
255 = reserved

The NC (Number of Characters) octet specifies the number of displayable characters to follow, and may take on values in the range 0 to 255 (decimal).
The C octet specifies the displayable character. The character shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9).

5.2.1.22 Send Prompt (16)
The format of this message shall be as shown in Figure 49/MTS22-C.
This command shall be used by the system to send and immediately display a prompt which is not pre-stored in a terminal's internal table of
prompts.

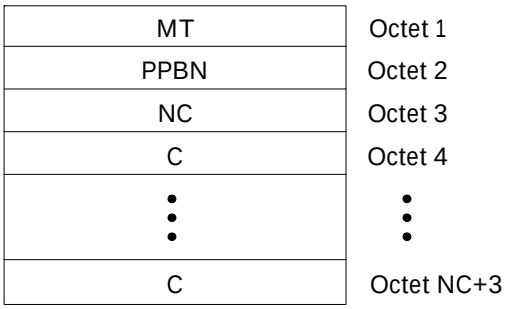


Figure 49/MTS22-C
Send Prompt command format
The PPBN octet's Prompt Position and Buffer Number fields specify the prompt area and buffer number to be affected by this command. Refer to § 5.2.1.1.4 for a detailed description of the PPBN octet.
The NC (Number of Characters) octet specifies the number of displayable characters to follow. The NC octet can take on values in the range 0 to 255 (decimal).
The C octet specifies the displayable character. The character shall be encoded in accordance with the extended ASCII format (see § 5.2.1.1.9).
Note - If the number of characters is too long for the terminal's prompt area, then the text will be truncated.

5.2.1.23 Deactivate prompt (17)
The format of this message shall be as shown in Figure 50/MTS22-C.
This command shall be used by the system to deactivate a specified prompt. The prompt text area and any associated delimiters and boxes shall also be blanked.

Note - Use of this command would, for example, make the display field available for text display via a Send String command (see § 5.2.1.2). In order to re-display a prompt and its delimiters and boxes, a Display Prompt (see § 5.2.1.20) or Send Prompt command shall be used.

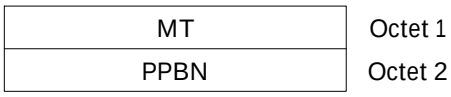


Figure 50/MTS22-C

Deactivate Prompt command format

The PPBN octet's Prompt Position and Buffer Number fields specify the affected prompt and buffer. Refer to § 5.2.1.1.4 for a detailed description of the PPBN octet.

5.2.1.24 Display Buffer (18)

The format of this command shall be as shown in Figure 51/MTS22-C.
This command shall be used by the system to indicate to a terminal which internal buffer is to be displayed.

Note - An example of the use of this command would occur when a terminal is switching between applications.

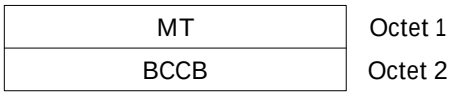


Figure 51/MTS22-C

Display Buffer command format

The BCCB octet's Buffer Number field specifies the affected buffer. No other BCCB parameters have meaning for this command. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.

5.2.1.25 Set LCD Indicator (19)

The format of this command shall be as shown in Figure 52/MTS22-C.
This command shall be used by the system to specify the state of a terminal's indicators.
The CRSR octet specifies the circle and square rates. Refer to § 5.2.1.1.5 for a detailed description of the CRSR octet.

Note - For LCD indicators that provide both a circle and a square segment, both CRSR octet parameters shall be used. For LED indicators (such as on a console) or a message waiting lamp (such as on a telephone set), only the circle parameter shall be used.

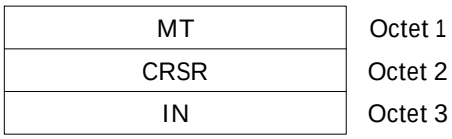


Figure 52/MTS22-C

Set LCD Indicator command format

The IN (Indicator Number) octet specifies the indicator number, and may take on values (decimal) as follows:

1) For set LCD indicators -
   0 to 251 = LCD indicator number
   252 = no indicator (this value has no effect on any indicators)
   253 = all indicators
   254 = microphone LED
   255 = message waiting lamp

2) For console LED indicators -
   0 = reserved
   1 to 7 = LED number

Note - LED number 1 on a console is in the leftmost position.

5.2.1.26 Load Flash Rate (1A)

The format of this command shall be as shown in Figure 53/MTS22-C.

This command shall be used by the system to load LCD flash rates into a terminal's memory.

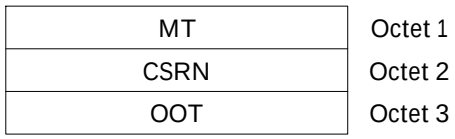


Figure 53/MTS22-C

Load Flash Rate command format

The CSRN octet defines the circle / square rate number. Refer to § 5.2.1.1.6 for a detailed description of the CSRN octet.

The OOT (ON / OFF Time) octet is defined as follows:

Bits 8 to 5 = ON time

Bits 4 to 1 = OFF time

The ON and OFF time parameters may take values in the range 0 to 15 (decimal). The values define multiples of 50 ms.

5.2.1.27 Send Signal (1B)

The format of this command shall be as shown in Figure 54/MTS22-C.

This command shall be used by the system to control various hardware functions in a terminal.

Note - This command is intended to be superseded in future applications by the Setup command (see § 5.2.1.46).

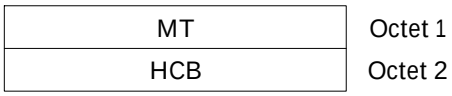


Figure 54/MTS22-C

Send Signal command format

The HCB (Hardware Control Byte) octet may take on either the HCB0 or HCB1 format, as defined in § 5.2.1.1.7.

5.2.1.28 Ringer Cadence High Continuous (1C)

The format of this command shall be as shown in Figure 55/MTS22-C.

This command shall be used by the system to start a terminal's ringer with high resolution cadence. The ringer shall repeat the cadence pattern continuously until turned off by a Send Signal command with the RNG (Ringer) bit turned off.

Note 1- This command shall be used with consoles only. Ringer Cadence commands are the only method by which console ringers can be activated.

The two CPB octets define the ringer cadence pattern. Octet 2 shall represent the higher-order octet. Each CPB octet bit shall represent a 100 ms time period. Refer to § 5.2.1.1.8 for a detailed description of the CPB octet.

Note - A subsequent cadence command shall immediately supersede the pattern currently in use.

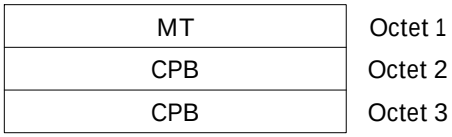
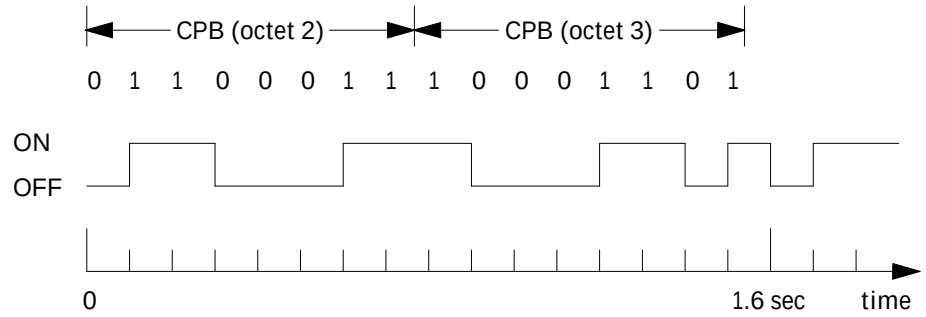


Figure 55/MTS22-C

Ringer Cadence High Continuous command format
Figure 56/MTS22-C illustrates an example ringer cadence pattern.



Note - each time division = 100 ms

Figure 56/MTS22-C
Ringer cadence example

5.2.1.29 Ringer Cadence High Pulsed (1D)
The format, coding and usage of this command shall be as defined in § 5.2.1.28.
This command shall be used by the system to start a terminal's ringer with high resolution cadence. Only one cadence cycle shall be generated. Each CPB octet bit shall represent a 100 ms time period.

5.2.1.30 Ringer Cadence Low Continuous (1E)
The format of this command shall be as shown in Figure 57/MTS22-C.
This command shall be used by the system to start a terminal's ringer with low resolution cadence. The ringer shall repeat the cadence pattern continuously until turned off by a Send Signal command with the RNG (Ringer) bit turned
off.

MT	Octet 1
CPB	Octet 2

Figure 57/MTS22-C
Ringer Cadence Low Continuous command format
The usage of this command shall be as defined in § 5.2.1.28, with the exception that only a single CPB octet shall be used. Each CPB octet bit shall represent a 200 ms time period.
Note - A subsequent cadence command shall immediately supersede the pattern currently in use.

5.2.1.31 Ringer Cadence Low Pulsed (1F)
The format and coding of this command shall be as defined in § 5.2.1.30.
This command shall be used by the system to start a terminal's ringer with low resolution cadence. Only one cadence cycle shall be generated.
The usage of this command shall be as defined in § 5.2.1.28, with the exception that only a single CPB octet shall be used. Each CPB octet bit shall represent a 200 ms time period.

5.2.1.32 Send DTMF (20)
The format of this command shall be as shown in Figure 58/MTS22-C.
This command is used by the system to instruct a terminal to generate DTMF tones.

MT	Octet 1
DIG12	Octet 2
DIG34	Octet 3

Figure 58/MTS22-C
Send DTMF command format

Note - This command does not employ a SAB octet.

The DIG12 and DIG34 octets each define pairs of DTMF digits as follows:

DIG12 octet: bits 8 to 5 = digit 1
bits 4 to 1 = digit 2
DIG34 octet: bits 8 to 5 = digit 3
bits 4 to 1 = digit 4

Each four-bit digit may take on the following values (decimal):
0 to 14 = valid DTMF digit
15 = null digit

Each digit generated shall persist for 100 ms. Successive digits shall be separated by 100 ms of silence. The null digit shall generate 100 ms of silence.

For each Send DTMF command received, the terminal shall reply with an Acknowledge DTMF response (see § 5.2.2.5). The system should wait for the response before issuing another Send DTMF command.

Note - If a keypad digit is pressed by the user while a terminal is processing a Send DTMF command, the command will be aborted and no response to the command will be sent.

5.2.1.33 Request Cursor Position (22)
The format of this command shall be as shown in Figure 59/MTS22-C.
This command is used by the system to request a terminal to report the Y,X coordinates of a specified cursor.

MT	Octet 1
BCCB	Octet 2

Figure 59/MTS22-C
Request Cursor Position command format
The BCCB octet specifies the desired cursor number. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.

5.2.1.34 Start Precise Ringer Cadencing (23)
The format of this command shall be as shown in Figure 60/MTS22-C.
This command shall be used by the system to activate the ringing sequence on a voice set.

MT	Octet 1
CPB	Octet 2
RP	Octet 3
CPP	Octet 4

Figure 60/MTS22-C
Start Precise Ringer Cadencing command format
The CPB octet specifies the cadence pattern to be used. Refer to § 5.2.1.1.8 for a detailed description of the CPB octet.

The RP (Ringing Period) octet specifies whether the ringing cycle shall be continuous or single-cycle, and also specifies the period of the ringing cycle. The RP octet is defined as follows:
Bit 8 = P (Pattern)
0 = continuous
1 = pulse

Bits 7 to 1 represent 8 (representing the duration of the variable CPB period), plus the number of 200 ms steps remaining in the total cadence period. To calculate the value of RP, the following procedure shall be used:
1) Choose ringer ON and ringer OFF according to desired period of the cadence pattern. These values will constitute the CPP.
2) Determine which bits of CPB are ones and which are zeros.
3) Count the number of ones in CPB (ONES) and the number of zeros (ZEROS).
4) Convert the Desired Cadence Period (DCP) to a value in milliseconds.
5) Calculate RP using the following

<Math?>

formula:
If RP is an integer value, the period of the cadence pattern is exactly matched.
The Cadence Pattern Parameters (CPP) octet defines the ringer cadencing ON/OFF time intervals. The ringer ON and ringer OFF values are 4-bit values (integer 0 to 15) representing a time interval in 50 ms steps from 0 ms to 750 ms. The default value is 4 (4 * 50 ms = 200 ms). The four most significant bits of the CPP define the ON time interval, while the four least significant bits define the OFF time interval.

Appendix I provides an example of how the RP, the CPB and the CPP values are assigned.

5.2.1.35 Set CODEC Gain (24)
The format of this command shall be as shown in Figure 61/MTS22-C.
This command shall be used by the system to set a terminal's transmit and receive gain.

MT	Octet 1
GAIN	Octet 2

Figure 61/MTS22-C
Set CODEC Gain command format

Note - This command does not employ a SAB octet.

The GAIN octet specifies the transmit and receive gain settings, and is defined as follows:

bit 8 = codec indicator
Value 0 = primary (default) codec
Value 1 = secondary codec
bits 7 to 5 = transmit gain setting
bit 4 = unused
bits 3 to 1 = receive gain setting

Transmit gain and receive gain may take on values in the range 0 to 7. The actual gain related to each value is a function of the type of set or console.

5.2.1.36 Start Voice Set Ringer (26)

The format of this command shall be as shown in Figure 62/MTS22-C.
This command shall be used by the system to activate the ringing sequence on a voice set.

MT	Octet 1
CPB	Octet 2
RP	Octet 3

Figure 62/MTS22-C
Start Voice Set Ringer command format

The CPB octet specifies the cadence pattern to be used. Refer to § 5.2.1.1.8 for a detailed description of the CPB octet. Each CPB octet bit shall represent a 200 ms time period.

The RP (Ringing Period) octet specifies whether the ringing cycle shall be continuous or single-cycle, and also specifies the period of the ringing cycle. The RP octet is defined as follows:

Bit 8 = P (Pattern)
0 = continuous
1 = pulse

Bits 7 to 1 define the ringing cycle period in multiples of 200 ms. The bits may take on values in the range 0 to 127 (decimal).

Note - The pattern defined by the CPB (see § 5.2.1.1.8) lasts 1.6 seconds, however the period value defined in the RP octet may be more or less than 1.6 seconds. If the RP octet's period value is less than 1.6 seconds, then the ringing cycle will be truncated. If the RP octet's period value is longer than 1.6 seconds in the case of continuous ringing, then the ringer will remain off from the end of the pattern until the beginning of a new period.

5.2.1.37 Define Initial Conditions (27)

The format of this command shall be as shown in Figure 63/MTS22-C.
This command shall be used by the system to initialize operational parameters after a terminal powers up or after a system reset.

MT	Octet 1
PAR1	Octet 2
PAR2	Octet 3

Figure 63/MTS22-C
Define Initial Conditions command format

Note - This command does not employ a SAB octet.

The PAR1 octet is defined as follows:
Bits 8 to 5 = ringing combination (0 to 15 decimal)
Bits 4 to 1 = ringing volume (0 to 15 decimal)
The PAR2 octet is defined as follows:
Bits 8 to 5 = music volume (0 to 15 decimal)
Bits 4 to 1 = handsfree volume (0 to 15 decimal)

Note - The system will use default values for a terminal which has not yet established a database.

5.2.1.38 Define Audio Mode (28)
The format of this command shall be as shown in Figure 64/MTS22-C.
This command shall be used by the system to place a terminal in one of the audio modes described below.

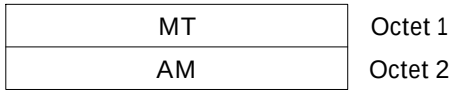


Figure 64/MTS22-C
Define Audio Mode command format

Note - This command does not employ an SAB octet.

The AM (Audio Mode) octet is defined as follows:
Bits 8 to 3 = unused

b2	b1	= audio mode
0	0	no mode (note 1)
0	1	music mode
1	0	ringer demo mode (note 2)
1	1	not used

Note 1- This mode implies a disable of the current audio mode. While in this mode, the volume keys may apply to the ringing volume or handsfree volume.
Note 2- In this mode, the terminal will generate an audible ringing pattern for the purpose of ringing combination selection.

5.2.1.39 Send Data Module (29)
The format of this command shall be as shown in Figure 65/MTS22-C.
This command shall be used by the system to respond to upstroke or downstroke of data module function keys Attention and Disconnect.

When the system receives notification of an Attention key or Disconnect key downstroke, it shall respond with the Attention or Disconnect mode of the Send Data Module command, respectively.
When the system receives notification of an Attention key or Disconnect key upstroke, it shall respond with the Idle mode of the Send Data Module command for BOTH
keys.

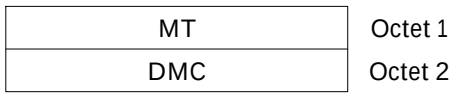


Figure 65/MTS22-C
Send Data Module command format

Note - This command does not employ an SAB octet.

The DMC (Data Module Command) octet is defined as follows:
Bits 8 to 3 = unused

b2 = D (Disconnect key)
0 idle
1 disconnect

b1 = A (Attention key)
0 idle
1 attention

5.2.1.40 Delete Character (2A)

This command has two possible formats, as shown in Figure 66/MTS22-C.
This command shall be used by the system to move a specified cursor back one position, and blank the character previously at that position. The following sequence of operations are performed:

1) acquire the specified cursor;
2) move the cursor back one position
3) blank the character at that position; and
4) return the cursor (as prior to the operation), if requested.

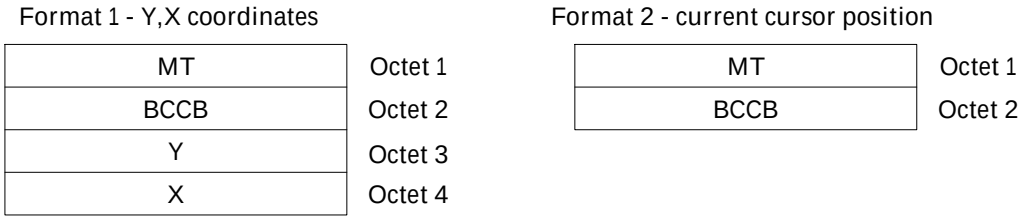


Figure 66/MTS22-C
Delete Character command format

All variations of the BCCB octet are allowed in this command. For example, if the BCCB octet's Return Cursor bit is set to 1, the specified cursor will appear to remain in place and the preceding character will be blanked.

5.2.1.41 Cancel Delay Update Of A Field (2B)

The format of this command shall be as shown in Figure 67/MTS22-C.
This command shall be used by the system to cancel any pending delayed update of a field for a specified buffer.

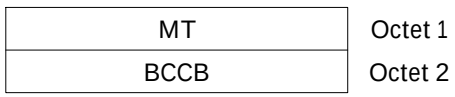


Figure 74/MTS22-C
Cancel Delay Update Of A Field command format

The BCCB octet's Buffer Number field specifies the buffer to be affected. No other BCCB octet fields have meaning for this command. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.

Note - The BCCB octet Buffer Number value shall not be equal to 0.

5.2.1.42 Initialize Display (2C)

The format of this command shall be as shown in Figure 68/MTS22-C.
This command shall be used by the system to initialize the display of a terminal which has internally-defined prompts. The following sequence of operations shall be performed:

1) clear the designated text lines in the specified buffer;
2) return the specified cursor to the “home” position; and
3) cancel any delayed updates for the specified buffer.

Note - The designated prompt lines are unaffected. This is in contrast to the Clear Screen command (see § 5.2.1.18), where all lines are affected.

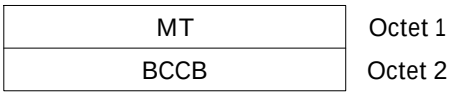


Figure 68/MTS22-C

Initialize Display command format

The BCCB octet specifies the desired buffer number and cursor number. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.

Note - The BCCB octet Buffer Number value shall not be equal to 0.

5.2.1.43 Select Stored Prompt (2D)

The format of this command shall be as shown in Figure 69/MTS22-C.

This command operates in a manner similar to the Display Prompt command (see § 5.2.1.20), with the exception that one extra parameter (First Prompt Position) is added.

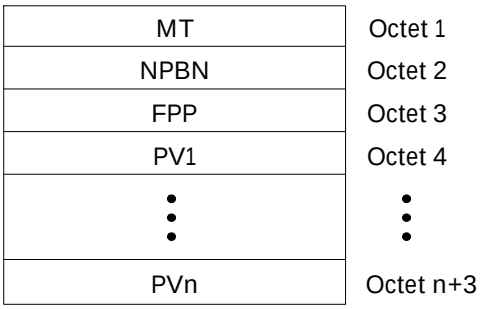


Figure 69/MTS22-C

Select Stored Prompt command format

The NPBN octet specifies the number of prompts and buffer number to be affected. The number of prompts is limited by the NPBN octet to 31 (decimal) per command. Refer to § 5.2.1.1.3 for a detailed description of the NPBN octet.

The FPP (First Prompt Position) octet specifies the first prompt position from which the PV octets are to be based.

The PV (Prompt Value) octets (1 to n) specify the individual prompt values. The prompts shall be listed in the order of actual display position on the terminal. The PV octet may take on the following values (decimal):

0 to 252 = valid prompt values
253 = deactivate prompt
254 = blank prompt position
255 = no change to prompt position

Note - For PV value 254, only the prompt text shall be blanked. The associated delimiters and boxes shall remain unaffected.

5.2.1.44 Define 5x7 Special Character (2E)

The format of this command shall be as shown in Figure 70/MTS22-C.

This command is used to define special characters using a 5x7 dot matrix.

Note - This command is superseded in newer terminals by the Define Character Pattern command (see § 5.2.1.45).

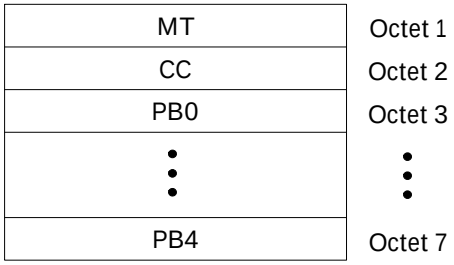
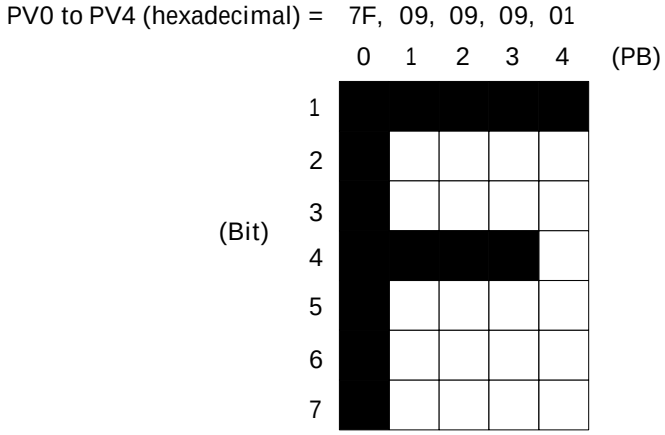


Figure 70/MTS22-C
Define 5x7 Special Character command format

Note - This command does not employ an SAB octet.

The CC (Character Code) octet defines one of 64 (decimal) special characters (see § 5.2.1.1.9).
The five PB (Pattern Byte) octets define the 5 columns in the 5x7 dot matrix. Bits 8 to 2 of each PB octet define which of the 7 dots in each column are activated. Refer to Figure 71/MTS22-C for a command coding



example.
Figure 71/MTS22-C
Define 5x7 Special Character example

5.2.1.45 Define Character Pattern (30)
The format of this command shall be as shown in Figure 72/MTS22-C.
This command operates in a manner similar to Define 5x7 Special Character (see § 5.2.1.44), with the exception that characters of variable height and width may be

defined.

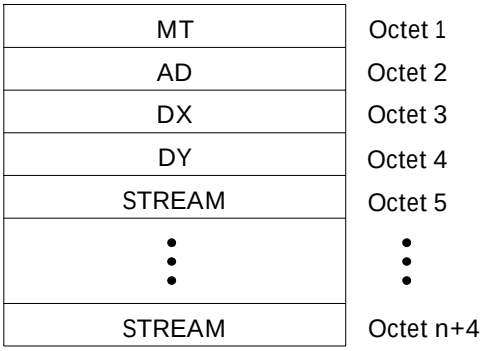


Figure 72/MTS22-C
Define Character Pattern command format

Note - This command does not employ an SAB octet.

The AD (ASCII Definition) octet defines one of 64 (decimal) special characters (see § 5.2.1.1.9).

The DX (Dimension X) octet defines the X dimension of the special character in pixels, and may take on values in the range 1 to 255 (decimal).

The DY (Dimension Y) octet defines the Y dimension of the special character in pixels, and may take on values in the range 1 to 255 (decimal).

The n STREAM octets define bit patterns for the pixels. For example, a character composed of 8x8 pixels (DX = DY = 8) could be defined by a STREAM field of 8 octets in length.

Note - The order of packing and rotation of the STREAM field does not form part of this MTS.

5.2.1.46 Setup (33)
The format of this command shall be as shown in Figure 73/MTS22-C.
This command shall be used by the system to control overall functions in a terminal.

Note - For new applications, this command is intended to supersede the Send Signal command (see § 5.2.1.27).

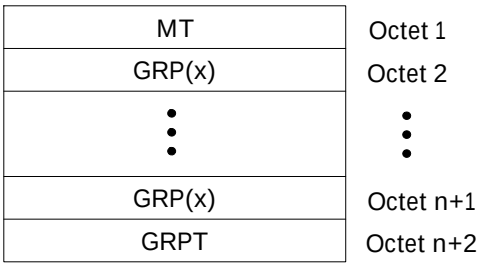


Figure 73/MTS22-C
Setup command format

Note - This command does not employ an SAB octet.

The formats and definitions of the GRP octets, composed of GRP0 to GRP7 and GRPT are defined in §§ 5.2.1.46.1 to 5.2.1.46.7. An example of the Setup command is shown in § 5.2.1.46.8. Proper sequencing of the GRP octets within the Setup command is essential to guarantee proper state transitions of the affected terminal.

5.2.1.46.1 GRP octet general format

The general format of a GRP octet is shown below:

b8	b7	b6	b5	b4	b3	b2	b1
F3	F2	F1	F0	G2	G1	G0	ON/OFF

Bits 8 to 5 (F3 to F0) specify the functions to be controlled. The functions are collected into groups, specified by bits 4 to 2 (G2 to G0). Each value of the group field shall specify an object, which in turn is composed of a set of functions. The current allocations of the group field are shown below:

G2	G1	G0	Object
0	0	0	handset
0	0	1	remote handsfree
0	1	0	DTMF generator
0	1	1	ringer
1	0	0	alternate codec
1	0	1	unassigned
1	1	0	unassigned
1	1	1	miscellaneous (unassigned)

Note - In the event that a new group (object) is created and does not contain many functions, it is recommended that they reside in GRP7 (miscellaneous).

Functions whose corresponding bits are set to 1 are activated or deactivated, depending on the state of the ON/OFF bit. Function bits which are set to 0 are unaffected by the operation. The state of the ON/OFF bit shall be 0 for OFF or 1 for ON. Unassigned Function bits should be set to 0.

5.2.1.46.2 GRP0 octet definition

The format of the GRP0 octet shall be as follows:

b8	b7	b6	b5	b4	b3	b2	b1
SIDE	HSNV	HSS	HSM	0	0	0	ON/OFF

The function bits are defined as follows:

SIDE = sidetone
HSNV = handset nominal volume (restore)
HSS = handset speaker
HSM = handset microphone

5.2.1.46.3 GRP1 octet definition

The format of the GRP1 octet shall be as follows:

b8	b7	b6	b5	b4	b3	b2	b1
LPF	HFA	RSS	RSM	0	0	1	ON/OFF

The function bits are defined as follows:

LPF = low-pass (noise) filter
HFA = handsfree algorithm
RSS = set speaker
RSM = set microphone

5.2.1.46.4 GRP2 octet definition

The format of the GRP2 octet shall be as follows:

b8	b7	b6	b5	b4	b3	b2	b1

x	x	DTRX	DTTX	0	1	0	ON/OFF

The function bits are defined as follows:
x = unassigned
DTRX = DTMF receive (to local transducers)
DTTX = DTMF transmit (to system via PCM)

5.2.1.46.5 GRP3 octet definition

The format of the GRP3 octet shall be as follows:

b8	b7	b6	b5	b4	b3	b2	b1
x	x	x	RING	0	1	1	ON/OFF

The function bits are defined as follows:
x = unassigned
RING = ringer

5.2.1.46.6 GRP4 octet definition

The format of the GRP4 octet shall be as follows:

b8	b7	b6	b5	b4	b3	b2	b1
ACLPF	ALTC	ACS	ACM	1	0	0	ON/OFF

The function bits are defined as follows:
ACLPF = low pass filter
ALTC = alternate codec enable
ACS = alternate codec speaker
ACM = alternate codec mic

5.2.1.46.7 GRPT octet definition

The GRPT (Group Termination) octet shall be used to specify the end of the sequence of GRP octets. The value of the GRPT octet shall be 0D (hexadecimal).

Note - The GRPT octet can be considered as a command, composed of an all-zero 4-bit function field and a 4-bit command field. At the present time, only the command field D (hexadecimal) has been allocated. The GRPT octet could be used to create 15 (decimal) other commands.

5.2.1.46.8 Setup command example

An example setup command is shown below for clarification:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	1	1	0	0	1	1	Setup MT code = 33
0	0	0	1	0	1	1	0	turn off ringer (GRP3)
0	0	1	1	0	1	0	1	allow DTMF both ways (GRP2)
1	0	1	1	0	0	0	1	turn on handset and sidetone (GRP0)
1	1	1	1	0	0	1	0	cancel any transducers (GRP1)
0	0	0	0	1	1	0	1	GRPT octet = 0D

5.2.1.47 Start New Call Tone (34)

The format of this command shall be as shown in Figure 74/MTS22-C.
This command shall be used by the system to activate the ringing sequence on a voice set.

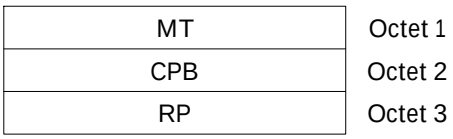


Figure 74/MTS22-C
Start New Call Tone command format

The CPB octet specifies the cadence pattern to be used. Refer to § 5.2.1.1.8 for a detailed description of the CPB octet. Each CPB octet bit shall represent a 200 ms time period.

The RP (Ringing Period) octet specifies whether the ringing cycle shall be continuous or single-cycle, and also specifies the period of the ringing cycle. The RP octet is defined as follows:
Bit 8 = P (Pattern)
0 = continuous
1 = pulse
Bits 7 to 1 define the ringing cycle period in multiples of 200 ms. The bits may take on values in the range 0 to 127 (decimal).

Note - The pattern defined by the CPB (see § 5.2.1.1.8) lasts 1.6 seconds, however the period value defined in the RP octet may be more or less than 1.6 seconds. If the RP octet's period value is less than 1.6 seconds, then the ringing cycle will be truncated. If the RP octet's period value is longer than 1.6 seconds in the case of continuous ringing, then the ringer will remain off from the end of the pattern until the beginning of a new period.
Note - This message is identical to the Start Voice Set Ringer (see § 5.2.1.36).

5.2.1.48 Wireless (35)

The format of this command shall be as shown in Figure 75/MTS22-C.
This command shall be used by the system to request a series of actions related to operation of wireless and cordless
sets.

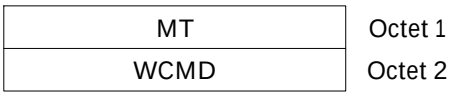


Figure 75/MTS22-C
Wireless command format
The WCMD octet specifies the specific wireless command requested by the system. The WCMD octet is defined as follows:
0 = reserved
1 = talk enable
2 = update display
3 to 255 = not used

5.2.1.48 Map Character To Unicode (36)
The format of this command shall be as shown in Figure 76/MTS22-C.

This message will be used by a system to map a non standard character code to a unicode value. This will permit systems to send the same text strings to all sets. Currently, some sets such as the 5010/5020 support the definition of non English characters in bitmap form. These bitmaps cannot be applied to sets such as the Webset and PDA phone which have different sized fonts and higher resolution displays.

The system will use the same character code that is used currently for a bitmap on a 510/5020, and specify which Unicode value the set should display when it receives a character in a text message.

Sets will not contain the entire Unicode character set, and will reject a message which tries to map to a value which the set does not support

Sets currently contain a number of predefined characters. New sets will contain these characters to maintain backwards compatibility. It will be possible to overwrite these predefined characters using the new Minet. (i.e. if you want $A3 to be the British Pound sign instead of i_acute, it can be remapped)

The message will be four bytes long.

MT	Oct
Character Code	Oct
Unicode Value (Upper)	Oct
Unicode Value (Lower)	Oct

Figure 76/MTS22-C

5.2.1.49 Go To URL (50)

Unlike most Minet messages, the Go To URL message is structure-based, rather than character-based. That is, the first byte of the message is the MT code, and the rest of the message is described by the following structure:

```
typedef struct {
        uint32_t        destBuffer;
        uint32_t        countUrl;
        uint8_t         Url[100];
} GO_TO_URL_MSG;
```

This command shall be used by the system to request the Webset to get the data from a URL and save it to one of its internal display buffers. The ‘destBuffer’ specifies the display buffer, ‘countUrl’ the size of the data, and ‘url[]’ the data.

5.2.1.50 Send Directory Number (51)

Unlike most Minet messages, the Send Directory Number message is structure-based, rather than character-based. That is, the first byte of the message is the MT code, and the rest of the message is described by the following structure:

```
typedef struct {
        uint32_t        countDn;
        uint8_t         directoryNumber[20];
} SEND_DIRECTORY_NUMBER_MSG;
```

This command shall be used by the system to upload the Webset’s Directory Number. The ‘countDn’ specifies the number of DN digits and ‘directoryNumber’ the DN digits.

5.2.1.51 Send Label String (52)

Unlike most Minet messages, the Send Label String message is structure-based, rather than character-based. That is, the first byte of the message is the MT code, and the rest of the message is described by the following structure:

```
typedef struct {
        uint32_t        destBuffer;
        uint32_t        labelLocation;
        uint32_t        labelType;
        uint32_t        labelCount;
        uint8_t         labelText[100];
} SEND_LABEL_STRING_MSG;
```

This command shall be used by the system to upload a soft label character string to the Webset. The ‘destBuffer’ specifies the buffer in which the label will be stored, ‘labelLocation’ the position of the label on the screen, ‘labelType’ the function of the key that got assigned by the system, ‘labelCount the number of characters in the label, and ‘labelText[]’ the label characters.

5.2.2 Responses from terminal to system

5.2.2.1 General format and definitions of elements

The general format for messages within the voice sets and consoles allocation block in the direction terminal to system shall be as shown in Figure 77/MTS22-C.

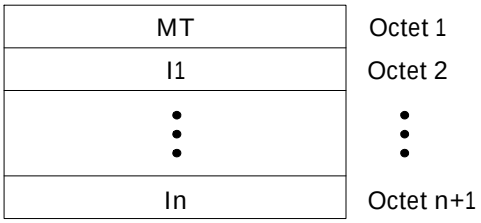


Figure 77/MTS22-C
Voice sets and consoles general response format

Note - The value of n is a function of the specific message type. Refer to the detailed descriptions following.

5.2.2.2 Keystrokes

Keystrokes shall be sent to the system according to the following rules:
1) for dial keys, report upstroke only;
2) volume keys shall be processed locally in the terminal;
3) QWERTY keyboard keys shall be processed as per applicable responses; and
4) for all other keys, report both downstroke and upstroke.

5.2.2.3 Send QWERTY Key (01)

The format of this response shall be as shown in Figure 78/MTS22-C.
This response shall be used by a terminal to report an upstroke or downstroke of a keyboard key.

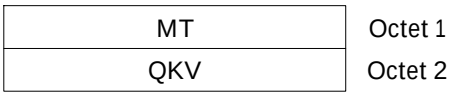


Figure 78/MTS22-C
Send QWERTY Key response format
The QKV (QWERTY Key Value) octet is defined as follows:

b8= U/D (Up/Down)
0 upstroke
1 downstroke

Bits 7 to 1 = standard ASCII characters (range 0 to 127 decimal), see § 5.2.1.1.9.

5.2.2.4 Send Scan Key (02)

The format of this response shall be as shown in Figure 79/MTS22-C.
This command shall be used by a terminal to report a downstroke or upstroke of a scan key.

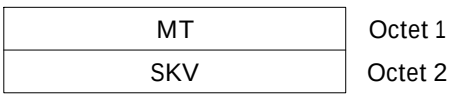


Figure 79/MTS22-C
Send Scan Key response format
The SKV (Scan Key Value) octet is defined as follows:

b8 = Set/Console
0 set
1 console
b7 = U/D (Up/Down)
0 upstroke
1 downstroke
Bits 6 to 1 = key value
Values 0 to 62 (decimal) = key code
Value 63 (decimal) = headset microphone flash

5.2.2.5 Acknowledge DTMF (03)

The format of this message shall be as shown in Figure 80/MTS22-C.
This response shall be sent by a terminal in reply to a Send DTMF command (see § 5.2.1.32). On receipt of this response, the system shall assume the terminal is ready to receive another Send DTMF command if necessary.

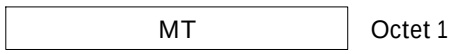


Figure 80/MTS22-C
Acknowledge DTMF response format

5.2.2.6 Report Cursor Position (04)

The format of this response shall be as shown in Figure 81/MTS22-C.
This command shall be used by a terminal to report a specified cursor's position in response to a Request Cursor Position command (see § 5.2.1.33).

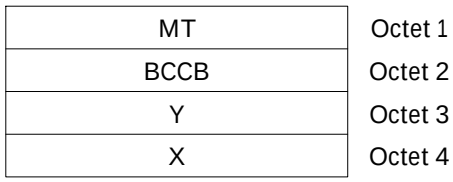


Figure 81/MTS22-C
Report Cursor Position response format
The BCCB octet specifies the cursor number for which the position is to be reported. Refer to § 5.2.1.1.1 for a detailed description of the BCCB octet.
The Y octet specifies the vertical line position, and may take on the following values (decimal):
0 = reserved
1 to 255 = valid line numbers
The X octet specifies the horizontal character position, and may take on the following values (decimal):
0 = reserved
1 to 255 = valid character numbers

5.2.2.7 Send Key Sequence (05)

The format of this response shall be as shown in Figure 82/MTS22-C.
This command shall be used by a terminal to report multiple key downstrokes and/or upstrokes.

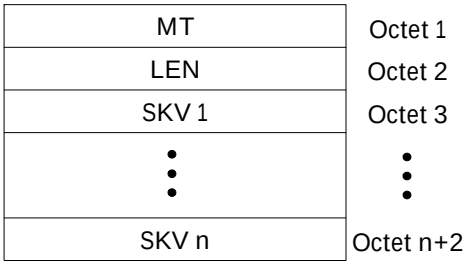


Figure 82/MTS22-C
Send Key Sequence response format
The following coding shall be used.
LEN = as of the number of SKV octets to follow (n)
SKV 1-n = Scan Key Value octet as defined in § 5.2.2.4
The LEN (Length) octet shall specify the number of SKV octets to follow. Valid values for LEN shall fall within the range 1 to the maximum send key sequence length, currently defaulted to 10. However, the maximum send key sequence length can be modified using the Local programmable key control command (see § 4.2.3.8)

5.2.2.8 Store Ring Combination (07)
The format of this response shall be as shown in Figure 83/MTS22-C.
This response shall be sent by a terminal whose ringing combination has been changed by the user.

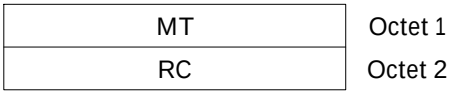


Figure 83/MTS22-C
Store Ring Combination response format

Note - This response does not employ an SAB octet.

The RC (Ringing Combination) octet is defined as follows:
Bits 8 to 5 = unused
Bits 4 to 1 = ringing combination (0 to 15 decimal)
The actual combinations are specific to a given terminal, however Table 7/MTS22-C lists the combinations available for the SS3DN, SS4DN and DNIC console.
Table 7/MTS22-C

Example ringing combination values

Ringing combination	Tones (Hz)	Pitch (low = 10 Hz, high = 16 Hz)
0	400 + 364	low
1	400 + 364	high
2	444 + 400	low
3	444 + 400	high
4	500 + 444	low
5	500 + 444	high
6	571 + 500	low
7	571 + 500	high
8	667 + 571	low
9	667 + 571	high
10	800 + 667	low
11	800 + 667	high
12	1000 + 800	low
13	1000 + 800	high
14	1333 + 1000	low
15	1333 + 1000	high

5.2.2.9 Store Handsfree Volume (08)

The format of this response shall be as shown in Figure 84/MTS22-C.
This response shall be sent by a terminal whose handsfree volume has been changed by the user.

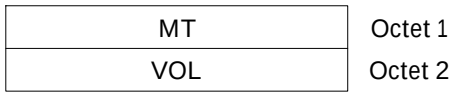


Figure 84/MTS22-C
Store Handsfree Volume response format

Note - This response does not employ an SAB octet.

The VOL (volume) octet is defined as follows:
Bits 8 to 5 = unused
Bits 4 to 1 = volume (0 to 15 decimal)

5.2.2.10 Store Ringing Volume (09)
The format and usage of this response shall be as defined in § 5.2.2.9.
This response shall be used by a terminal whose ringing volume has been changed by the user.

5.2.2.11 Store Music Volume (0A)
The format and usage of this response shall be as defined in § 5.2.2.9.

This response shall be used by a terminal whose music volume has been changed by the user.

5.2.3 Requests from terminal to system

5.2.3.1 Keepalive Poll (7E)

The format of this command shall be as shown in Figure 85/MTS22-C.

Figure 85/MTS22-C
This message is used by the terminal to check that the other end of the Minet link (i.e., the system) is still alive. A link layer Ack provides confirmation.

6 Data messages

In the message descriptions following, the MT (Message Type) codes are presented as hexadecimal values, and are shown in brackets following the message titles.

6.1 Allocated messages

Three main classes of messages are defined within the data sets allocation block as follows:

1) data transport;
2) DNIC data set; and
3) MCX (Mitel Coax) data sets.

Table 8/MTS22-C to Table 12/MTS22-C list all messages defined within the data sets allocation block. MT code values are shown in hexadecimal.

Table 8/MTS22-C
Data transport messages

MT	**Message**
80	ASCII Data
81	RS-232 Status + ASCII Data
85	SX-2000 Messaging
86	Reserved
87	Bularo Mode ASCII Data
89	Synchronous Data
8A-BE	Not used

Table 9/MTS22-C
DNIC data sets commands

MT	**Command (system to terminal)**
BF + 00	Not used
BF + 01	Request RS-232 Interface State
BF + 02	Request Link Layer Statistics
BF + 03	Specify Auto-baud Characters
BF + 04	Specify Attention Character
BF + 05	Specify EIA Signals
BF + 06	Specify Interface Options
BF + 07	Specify UART Parameters
BF + 08	Specify Inactivity Timers
BF + 09	Force Layer 3 Idle Mode
BF + 0B - BF + 0F	Not used
BF + 10	Terminate B Channel Session
BF + 11	Specify DS2200 Network Sync Timing
BF + 12	Specify Synchronous Operating Mode
BF + 13	Not used
BF + 14	Request DS2100 Switch Settings
BF + 15	Specify EIA Inputs Default ON
BF + 16	Not used
BF + 17	Request TX/RX Character Count
BF + 18	Not used
BF + 19	Specify EIA Outputs Default ON
BF + 1A	Specify Disconnect Lead
BF + 1B	Specify RTS/CTS Delay
BF + 1C	Request Packet Statistics
BF + 1D	Not used
BF + 1E	Clear Network Management Statistics
BF + 1F	Specify DS2200 Data Mask
BF + 20	Not used

BF + 21	Request Buffer Statistics
BF + 22	Specify Buffer / MiLAP Parameters
BF + 23	Specify Hardware Flow Control
BF + 24	Specify D Channel Options
BF + 25	Specify DTR Disconnect Timer
BF + 26	Specify Interface Type
BF + 27 - BF + FF	Not used

Table 10/MTS22-C
DNIC data sets messages

MT	**Command (terminal to system)**
BF + 00 - BF + 0A	Not used
BF + 0B	Report RS-232 Interface State
BF + 0C	Report Link Layer Statistics
BF + 0 D	Report Baud Rate Information
BF + 0E	Report B to T Reason
BF + 0F	Report Keystroke
BF + 10 - BF + 12	Not used
BF + 13	Report DS2200 Network Sync Timing
BF + 14, BF + 15	Not used
BF + 16	Report DS2100 Switch Settings
BF + 17	Not used
BF + 18	Report TX/RX Character Count
BF + 19 - BF + 1C	Not used
BF + 1D	Report Packet Statistics
BF + 1E, BF + 1F	Not used
BF + 20	Report Buffer Statistics
BF + 21 - BF + FF	Not used

Table 11/MTS22-C
MCX command

MT	**Message (controller to terminal)**
82	Command frame
83	Data Frame
84	General Frame
88	Write Extended Attribute Buffer Null

Table 12/MTS22-C
MCX response

MT	**Message (terminal to controller)**
82	Solicited Response
83	Unsolicited Response
84	Error Response

6.2 Message Descriptions

6.2.1 Data transport

6.2.1.1 Asynchronous transport

The MiNET asynchronous data transport protocol provides the following functions:

- EIA control signal pass-through in-band;
- minimum control signal and data delay;
- full word size, parity, stop bits and break support; and
- all major data rates, up to and including 19.2 kbit/s.

6.2.1.2 EIA interface status encoding

Certain data transport messages allow transmission of EIA interface status information. The EIA status octets shall be encoded as defined in this section.

6.2.1.2.1 EIA1 octet

The EIA1 octet shall have the following format:

b8	b7	b6	b5	b4	b3	b2	b1
EXTN	GENDER		RI	SPACE	CTS/RTS	DCD/CTS	RTS/DCD DTR/DSR

The EXTN (Extension) bit shall be set to 1 when the following octet also contains RS-232 interface status information.

The GENDER bit shall be encoded or interpreted according to the following definition:

GENDER	Direction	Meaning
0	Transmit	Attached local device = DTE
1	Transmit	Attached local device = DCE
0	Receive	Attached remote device = DTE
1	Receive	Attached remote device = DCE

Note - The GENDER bit refers to the attached device, rather than the data set.

The RI (Ring Indicate) bit relates to the RS-232 RI signal, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	RI = 0
DCE	Transmit	Set RI bit in accordance with RI signal
DTE	Receive	Drive RI signal in accordance with RI bit
DCE	Receive	Ignore

The SPACE bit relates to the all-spacing (break) RS-232 line state, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set SPACE bit in accordance with BREAK state
DCE	Transmit	Set SPACE bit in accordance with BREAK state
DTE	Receive	Send BREAK state in accordance with state of SPACE bit
DCE	Receive	Send BREAK state in accordance with state of SPACE bit

Note - SPACE bit = 1 indicates BREAK signal active.

The CTS/RTS (Clear To Send/Request To Send) bit relates to the RS-232 CTS and RTS signals, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set CTS/RTS bit in accordance with CTS signal
DCE	Transmit	Set CTS/RTS bit in accordance with RTS signal
DTE	Receive	No action
DCE	Receive	No action

Note - This bit is intended as a monitor function only.

The DCD/CTS (Data Carrier Detect/Clear To Send) bit relates to the RS-232 DCD and CTS signals, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set DCD/CTS bit in accordance with DCD signal
DCE	Transmit	Set DCD/CTS bit in accordance with CTS signal
DTE	Receive	Drive CTS signal in accordance with state of DCD/CTS bit
DCE	Receive	No action

The RTS/DCD (Request To Send/Data Carrier Detect) bit relates to the RS-232 RTS and DCD signals, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set RTS/DCD bit in accordance with RTS signal
DCE	Transmit	Set RTS/DCD bit in accordance with DCD signal
DTE	Receive	Drive DCD signal in accordance with state of RTS/DCD bit
DCE	Receive	Drive RTS signal in accordance with state of RTS/DCD bit

The DTR/DSR (Data Terminal Ready/Data Set Ready) bit relates to the RS-232 DTR and DSR signals, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set DTR/DSR bit in accordance with DTR signal
DCE	Transmit	Set DTR/DSR bit in accordance with DSR signal
DTE	Receive	Drive DSR signal in accordance with state of DTR/DSR bit
DCE	Receive	Drive DTR signal in accordance with state of DTR/DSR bit

6.2.1.2.2EIA2 octet

The EIA2 octet shall have the following format:

b8	b7	b6	b5	b4	b3	b2	b1
EXTN	0	0	0	RDL	ALB	RS	MI

The EXTN (extension) bit shall be set to 1 when the following octet also contains RS-232 interface status information.

The RDL (Remote Data Loopback) bit relates to the RS-232 RDL signal, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set RDL bit to 0
DCE	Transmit	Set RDL bit to 0
DTE	Receive	Drive RDL signal in accordance with state of RDL bit
DCE	Receive	Drive RDL signal in accordance with state of RDL bit

Note - RDL control is only available with the DS2100 data set using an attached Mitel Modem Adapter.

The ALB (Analog Loopback) bit relates to the RS-232 ALB signal, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set ALB bit to 0
DCE	Transmit	Set ALB bit to 0
DTE	Receive	Drive ALB signal in accordance with state of RDL bit
DCE	Receive	Drive ALB signal in accordance with state of RDL bit

Note - ALB control is only available with the DS2100 data set using an attached Mitel Modem Adapter.

The RS (Rate Select) bit relates to the RS-232 RS signal, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set RS bit to 0
DCE	Transmit	Set RS bit to 0
DTE	Receive	No action
DCE	Receive	Drive RS signal in accordance with state of RS bit

Note - RS control is only available with the DS2100 data set using an attached Mitel Modem Adapter.

The MI bit relates to the MI/MIC function in analog modems, and shall be encoded or interpreted according to the following definition:

GENDER	Direction	Action
DTE	Transmit	Set MI bit to 0
DCE	Transmit	Set MI bit to 0
DTE	Receive	No action
DCE	Receive	Send MI/MIC signal in accordance with state of MI bit

Note - MI/MIC control is only available with the DS2100 data set.

6.2.1.3 ASCII Data (80)

The format of this message shall be as shown in Figure 86/MTS22-C.
This message shall be used by a terminal to transmit up to 13 (decimal) octets of ASCII data to a remote application. The maximum supported terminal speed shall be 19.2 kbit/s.

This message may be used to transfer data from the attached device to the system via the D channel if so enabled via the Specify Interface Options command (see § 6.2.2.1.6). Similarly this message may also be used to transfer data from the D channel to the attached device if the proper EIA signals have been activated via the Specify EIA Signals command (see § 6.2.2.1.5). Normally, however, the primary use of this command is to transfer data via a B
channel.

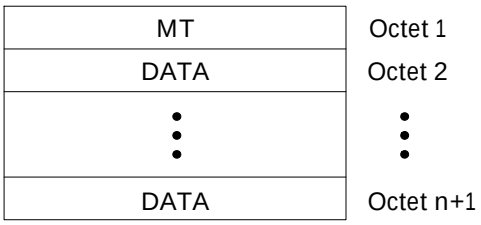


Figure 86/MTS22-C
ASCII Data message format
The maximum value of n shall be 13 (decimal).
The data octets shall be encoded using the standard ASCII character set, as defined by ANSI X3.4 [1].

6.2.1.4 RS-232 Status And ASCII Data (81)

The format of this message shall be as shown in Figure 87/MTS22-C.
This message shall be used by a terminal to transmit RS-232 interface status information to a remote application. Optionally, ASCII data may also be appended to the message, providing that the overall message size, MT code included, does not exceed 14 octets in length.
The maximum supported terminal speed shall be 19.2kbit/s.

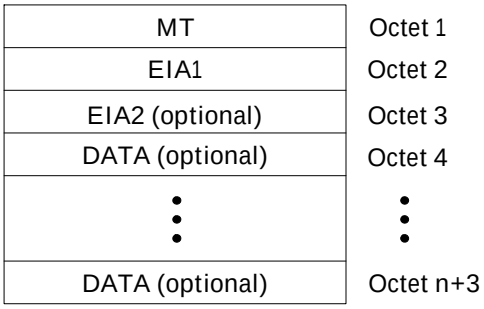


Figure 87/MTS22-C
RS-232 Status And ASCII Data message format
The maximum value of n shall be 11 (decimal) if the EIA2 octet is included, otherwise the maximum value of n shall be 12 (decimal).

The EIA1 and optional EIA2 octets shall be encoded in accordance with § 6.2.1.2. The EIA2 octet shall only be included if the extension bit in the EIA1 octet is set to 1.
The data octets shall be encoded using the standard ASCII character set, as defined by ANSI X3.4 [1].

6.2.1.5 SX-2000 Messaging (85)

The format of this message shall be as shown in Figure 88/MTS22-C.
This message is used to transfer 32 octets of data between the system and a Superset 7 console. Description of the contents of the data field is outside the scope of this specification.

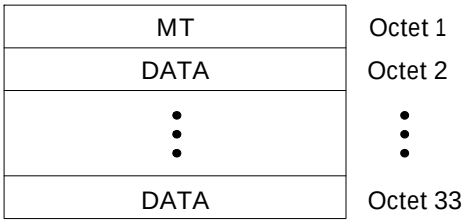


Figure 88/MTS22-C
SX-2000 Messaging message format

6.2.1.6 Bularo Mode ASCII Data (87)

The format of this message shall be as shown in Figure 89/MTS22-C. The overall message size, MT code included, shall not exceed 14 octets.The maximum supported terminal speed shall be 19.2 kbit/s.

Note - This message shall be used for communications between data sets and the SX-200D main control only.

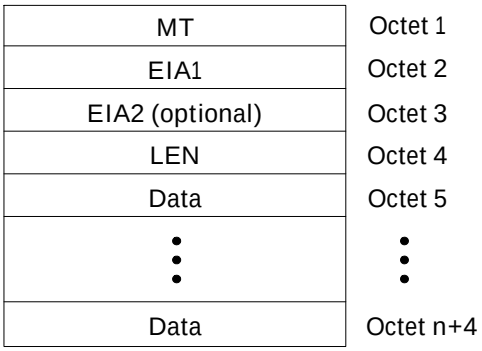


Figure 89/MTS22-C
Bularo Mode ASCII Data message format
The maximum value of n shall be 10 (decimal) if the EIA2 octet is present, otherwise the maximum value of n shall be 11 (decimal).

The EIA1 and optional EIA2 octets shall be encoded in accordance with § 6.2.1.2. The EIA2 octet shall only be included if the extension bit in the EIA1 octet is set to 1.

The LEN (Length) octet shall indicate the number of ASCII data octets to follow.
The overall length of the message shall constitute an even number of octets. In the event that the number of octets is odd, an extra data octet shall be added. The value of the LEN octet, however, shall not reflect the addition of the extra octet (i.e. the value of the LEN octet shall reflect the true number of data octets transferred).

The data octets shall be encoded using the standard ASCII character set, as defined by ANSI X3.4 [1].

6.2.1.7 Synchronous Data (89)

The format of this message shall be as shown in Figure 90/MTS22-C.
This message shall be used by a terminal to transmit synchronous data to a remote application. The maximum supported terminal speed shall be 19.2 kbit/s.
Note - The synchronous transport method described in this section is supported by the DS2200 data set.

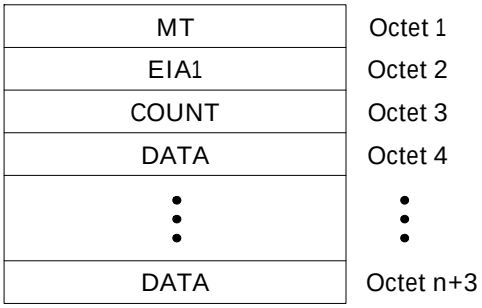


Figure 90/MTS22-C
Synchronous Data message format

The maximum value of n shall be 9 (decimal).
The EIA1 octet shall be encoded in accordance with § 6.2.1.2.1.

Note - The EIA1 octet's extension bit shall have no meaning in this message.

The COUNT octet shall contain a modulo 256 running total count of all data octets transmitted, up to and including the data octets of the current message.

Note - The COUNT octet is used for the purpose of maintaining end-to-end data rate synchronization. For a detailed description of the implementation method, refer to PS.3 [14].

The average size of the data field is generally a function of the data rate.
Table 13/MTS22-C lists average data field sizes for some of the most common baud rates.

Table 13/MTS22-C
Data field size versus baud rate

Baud rate (bit/s)	Average data field size (octets)
1200	2
2400	3
4800	4
9600	5
19200	6

Note - The transport of synchronous data at rates above 19.2 kbit/s (e.g. 48 kbit/s, 56 kbit/s and 64 kbit/s) shall not form part of this specification. Details of procedures for such higher speeds may be found in PS.3 [14].

The Synchronous Data messages shall be transferred to and from the data link layer via the DL-UNIT-DATA primitives, thereby utilizing the data link layer's unacknowledged information transfer service.

Note - Procedures to recover from lost or corrupted messages shall not form part of this specification. Such procedures would normally be the responsibility of the end terminal equipment.

6.2.2 DNIC data sets

Note - All messages in this section shall use an MT code consisting of two octets. The first octet shall always contain the extension code BF (hexadecimal).

6.2.2.1 Commands from system to terminal

6.2.2.1.1 Request RS-232 Interface State (BF+01)

The format of this message shall be as shown in Figure 91/MTS22-C.
This command shall be used by the system to request information regarding the terminal's RS-232 port.

MT	Octet 1
MT	Octet 2

Figure 91/MTS22-C
Request RS-232 Interface State command format

6.2.2.1.2 Request Link Layer Statistics (BF+02)

The format of this command shall be as shown in Figure 92/MTS22-C.
This command shall be used by the system to request any data link layer statistics supported by a terminal.

MT	Octet 1
MT	Octet 2

Figure 92/MTS22-C
Request Link Layer Statistics

6.2.2.1.3 Specify Auto-baud Characters (BF+03)

The format of this command shall be as shown in Figure 93/MTS22-C.
This command shall be used by the system to specify 1 to 12 auto-baud characters to be sent by the terminal to the attached device.

If no auto-baud characters are specified, the command shall disable the sending of any auto-baud characters. When the command has been received by a data set, the auto-baud characters shall always be sent to the attached device when the data set moves to a B channel.

MT	Octet 1
MT	Octet 2
ICTF	Octet 3
C (optional)	Octet 4
⋮	⋮
C (optional)	Octet n+3

Figure 93/MTS22-C
Specify Auto-baud Characters command format
The ICTF (Inter-Character Time Fill) octet specifies the approximate time between the start points of two consecutive auto-baud characters, and may take on values (decimal) as follows:

0 = no delay
1 = 0 to 10 ms delay
2 = 10 ms to 20 ms delay
to
127 = 1260 ms to 1270 ms delay

The inter-character time specified in this command shall also be applied after a data set has disconnected from the D channel and before the first auto-baud character is sent to the attached device.

The C octets specify the auto-baud characters to be sent. The character values shall be encoded as defined in ANSI X3.4 [1], and may take on values in the range 0 to 7F (hexadecimal).

6.2.2.1.4 Specify Attention Character (BF+04)

The format of this command shall be as shown in Figure 94/MTS22-C.
This command shall be used by the system to specify an attention character for a terminal. The terminal shall monitor the incoming stream from the attached device for the presence of the specified attention character. When the attention character is received, the terminal shall drop the B channel connection.

MT	Octet 1
MT	Octet 2
C (optional)	Octet 3

Figure 94/MTS22-C
Specify Attention Character command format

The C octet specifies the attention character to be used. The character value shall be encoded as defined in ANSI X3.4 [1]. If no C octet is present, all attention character monitoring shall be disabled.

6.2.2.1.5 Specify EIA Signals (BF+05)

The format of this command shall be as shown in Figure 95/MTS22-C.
This command shall be used by the system to specify the EIA signal states on a terminal's interface to its attached device. The terminal shall never change the signal states on its own.

Note - This command may also be used for interfaces other than RS-232 (e.g. V.35), however, the mapping of bits to interface signals shall not form part of this specification.

MT	Octet 1
MT	Octet 2
EIA1	Octet 3
EIA2 (optional)	Octet 4
EXT (optional)	Octet 5
⋮	⋮
EXT (optional)	Octet n+4

Figure 95/MTS22-C
Specify EIA Signals command format

The EIA1 and optional EIA2 octets shall be encoded in accordance with § 6.2.1.2. The EIA2 octet shall only be included if the extension bit in the EIA1 octet is set to 1.
The EXT (Extension) octets allow for the transfer of additional information. Specific encodings are for further study.

6.2.2.1.6 Specify Interface Options (BF+06)

The format of this message shall be as shown in Figure 96/MTS22-C.
This command shall be used by the system to specify to the terminal interface options for flow control, BREAK and inputs to be defaulted HIGH.

MT	Octet 1
MT	Octet 2
OPT	Octet 3
XON (optional)	Octet 4
XOFF (optional)	Octet 5

Figure 96/MTS22-C

Specify Interface Options command format

The OPT (Options) octet is defined as follows:

b8 b7 = Enable transmission of RS-232 receive data to D channel option
- 0 0 no change
- 0 1 option OFF
- 1 0 option ON
- 1 1 option OFF

b6 b5 = RS-232 force HIGH option
- 0 0 no change
- 0 1 option OFF
- 1 0 option ON
- 1 1 option OFF

Note - When the gender of the attached device is DTE, then the DTR and RTS inputs to the terminal are affected. When the gender of the attached device is DCE, then the DSR, CTS and DCD inputs to the terminal are affected.

b4 b3 = system attention on BREAK option
- 0 0 no change
- 0 1 transparent
- 1 0 system attention
- 1 1 transparent

Note - BREAK shall be defined as the reception by the terminal from the attached device of an RS-232 space condition with a duration of greater than 200 ms.

b2 b1= flow control option
- 0 0 no change
- 0 1 CTS
- 1 0 XON/XOFF
- 1 1 none

Note - When the XON/XOFF option is selected, then the next 2 octets of the command shall be present and shall specify the actual characters to be used. The characters which are downloaded in octets 4 and 5 need not contain the appropriate parity bit, since the terminal will set the parity bit in accordance with the parity specifies in the Specify UART Parameters command (see § 6.2.2.1.7). However, when transparent parity has been selected in the Specify UART Parameters command, the system shall ensure that the downloaded XON and XOFF characters contain a parity bit which will be acceptable to the attached device.

The XON and XOFF octets shall define the actual XON and XOFF characters to be used. The character values shall be encoded as defined in ANSI X3.4 [1], and may taken on values in the range 0 to 7F (hexadecimal).

6.2.2.1.7 Specify UART Parameters (BF+07)

The format of this command shall be as shown in Figure 97/MTS22-C.
This command shall be used by the system to configure terminal UART (Universal Asynchronous Receiver/Transmitter) parameters, such as baud rate, parity, number of stop bits, and character length.

MT	Octet 1
MT	Octet 2
BR	Octet 3
UPAR (optional)	Octet 4

Figure 97/MTS22-C

Specify UART Parameters command format

The BR (Baud Rate) octet specifies the baud rate, and may take on values (hexadecimal) as follows:

00 = no change
01 = 50 bit/s
02 = 75 bit/s
10 = 110 bit/s
11 = 134.5 bit/s
20 = 150 bit/s
30 = 200 bit/s
40 = 300 bit/s
50 = 600 bit/s
60 = 1200 bit/s
61 = 1800 bit/s
62 = 2000 bit/s
70 = 2400 bit/s
71 = 3600 bit/s
80 = 4800 bit/s
81 = 7200 bit/s
90 = 9600 bit/s
91 = 14400 bit/s
92 = 16600 bit/s
A0 = 19200 bit/s

The UPAR (UART Parameters) octet is defined as follows:

b8 b7 = number of stop bits
- 0 0 no change
- 0 1 1 stop bit
- 1 0 2 stop bits
- 1 1 not used

b6 b5 b4 = character length
- 0 0 0 no change
- 0 0 1 5 bits
- 0 1 0 6 bits
- 0 1 1 7 bits
- 1 0 0 8 bits
- 1 0 1 9 bits
- 1 1 0 not used
- 1 1 1 not used

b3 b2 b1 = parity

0	0	0	no change
0	0	1	mark
0	1	0	space
0	1	1	even
1	0	0	odd
1	0	1	none
1	1	0	not used
1	1	1	not used

6.2.2.1.8 Specify Inactivity Timers (BF+08)

The format of this command shall be as shown in Figure 98/MTS22-C.
This command shall be used by the system to set values for a terminal's inactivity disconnect timer and connect confirmation timer.

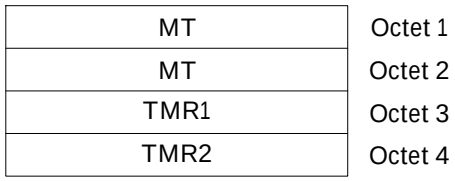


Figure 98/MTS22-C
Specify Inactivity Timers command format

The TMR1 (Timer 1) octet specifies the inactivity disconnect timer, while the TMR2 (Timer 2) octet specifies the connect confirmation timer. Each timer may take on values (decimal) as follows:

0 = disable timer
1 to 255 = 1 minute to 255 minutes

Note - The timer values have a +1 minute resolution. For example, a timer value of 1 represents a time range of 1 to 2 minutes.

When the inactivity timer is exceeded, the terminal shall perform a data link layer disconnect from the B channel, reconnect on the D channel, and inform the system of the time-out via the Report B To D Reason response (see § 6.2.2.2.4).

6.2.2.1.9 Force Layer 3 Idle Mode (BF+09)

The format of this command shall be as shown in Figure 99/MTS22-C.
This command shall be used by the system to force a terminal into the idle mode at layer 3.

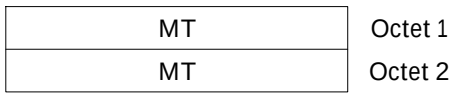


Figure 99/MTS22-C
Force Layer 3 Idle Mode command format

The reception of this command by a terminal shall not affect the state of the data link layer session, nor shall it change the state of any downloaded parameters. The only exception to this occurs with the enable transmission of RS-232 receive data to D channel option in the Specify Interface Options command (see § 6.2.2.1.6). This option shall always be set to OFF when the terminal is in the idle state.

6.2.2.1.10 Start B Channel (BF+0A)

The format of this command shall be as shown in Figure 100/MTS22-C.
This command shall be used by the system to set various B channel operation modes in a terminal.

MT	Octet 1
MT	Octet 2
MODE1	Octet 3

Figure 100/MTS22-C

Start B Channel command format

The MODE1 octet shall be defined follows:

Bit 8 = not used

b7 b6 = operating mode

b7	b6	
0	0	normal (endpoint to endpoint)
0	1	master multipoint
1	0	slave multipoint
1	1	not used

Note1 - Normal operating mode indicates that only a single data terminal is present at each end of a circuit switch connection.

Note2 - Master multipoint mode applies to a data terminal whose transmitter connects to the receivers of multiple other devices, and whose receiver is multiplexed over the other devices' transmitters.

Note3 - Slave multipoint mode applies to data terminals whose receivers are connected to the transmitter of a master terminal, and whose transmitters are alternately connected to the receiver of the master terminal.

b5 b4 b3 b2 b1 = ANS/ORG/GoA mode

b5	b4	b3	b2	b1		
x	x	x	0	0	ANS (answer mode)	
x	x	x	0	1	ORG (originate mode)	
x	x	x	1	0	GoA (go-ahead mode)	
x	x	1	x	x	execute downloaded code	
x	1	x	x	x	clear input buffer	
1	x	x	x	x	Bularo mode	

Note1 - Answer mode indicates that the data terminal shall connect to a B channel as a destination device. Its behaviour shall then be the same as an originate mode data terminal.

Note2 - Originate mode indicates that the data terminal shall connect to a B channel and initiate the establishment of a new session. It is allowable for both endpoints of a data call to connect to a B channel in originate mode.

Note3 - Go-ahead mode indicates that the data terminal shall obey the go-ahead protocol (see MTS20 [9]).

Note4 - The execute downloaded code option allows a data terminal to operate from ROM-based firmware while on the D channel, but to jump to RAM-based software for operation on the B channel.

Note5 - The clear input buffer option allows a data terminal to clear its input buffers (containing data from the attached device) prior to connection to the B channel.

Note6 - Bularo mode is used when a data terminal connects to a B channel for communication with the main control in SX-200D systems. In this mode, the data terminal shall use the data format as defined in § 6.2.1.6.

6.2.2.1.11 Terminate B Channel Session (BF+10)

The format of this command shall be as shown in Figure 101/MTS22-C.

This command shall be used by the system to cause a terminal to terminate its current B channel connection.

MT	Octet 1
MT	Octet 2

Figure 101/MTS22-C
Terminate B Channel Session command format
On receipt of this command, and termination from the B channel, the terminal shall reply with a Report B To D Reason response (see § 6.2.2.2.4).

6.2.2.1.12 Specify DS2200 Network Synchronization Timing (BF+11)

The format of this command shall be as shown in Figure 102/MTS22-C.
This command shall be used by the system to control DS2200 network synchronization functions.

MT	Octet 1
MT	Octet 2
NETSYNC MT	Octet 3
MAX SLIPS (optional)	Octet 4

Figure 102/MTS22-C
Specify DS2200 network synchronization timing command format
This command is actually composed of 4 sub-commands, determined by the value of the NETSYNC MT octet as follows:

NETSYNC MT	Sub-command
1	Disable Network Synchronization
2	Enable Network Synchronization
3	Request Long Term Accumulator
4	Request Clock Source

The sub-commands are defined in §§ 6.2.2.1.12.1 to 6.2.2.1.12.4.
The MAX SLIPS octet shall be used only with the Enable Network Synchronization sub-command (see § 6.2.2.1.12.2).

6.2.2.1.12.1 Disable Network Synchronization

NETSYNC MT = 1
This sub-command shall be used by the system to cause the terminal's network synchronization process to enter the DISABLED state. The terminal shall respond with an Acknowledge Network Synchronization Disable sub-response (see § 6.2.2.2.6.1).

6.2.2.1.12.2 Enable Network Synchronization

NETSYNC MT = 2
This sub-command shall be used by the system to cause the terminal's network synchronization process to enter the ACQUISITION state. The terminal shall respond with an Acknowledge Network Synchronization Enable sub-response (see § 6.2.2.2.6.2).
The MAX SLIPS octet shall be used to specify the maximum number of DS2200 buffer slips allowed prior to the terminal reporting a slip threshold occurrence. The default value of this octet shall be 1.

6.2.2.1.12.3 Request Long Term Accumulator

NETSYNC MT = 3
This sub-command shall be used by the system to cause the terminal's network synchronization process to enter the TRACKING state. The terminal shall respond with an Acknowledge Request Long Term Accumulator sub-response (see § 6.2.2.2.6.3).

6.2.2.1.12.4 Request Clock Source

NETSYNC MT = 4

This command shall be used by the system to query the terminal's ability to act as a synchronization source.
The terminal shall respond with a Report Clock Source sub-response (see § 6.2.2.2.6.4).
The terminal shall signal its ability to act as a synchronization source if all of the following conditions are met:

- the terminal is in network timing mode;
- the network clock is present from the interface; or
- the interface is in a ready state.

6.2.2.1.12.5 Specify Synchronous Operating Mode (BF+12)

The format of this command shall be as shown in Figure 103/MTS22-C.
This command shall be used by the system to specify the operating modes for a DS2100 dataset.

MT	Octet 1
MT	Octet 2
SPEED	Octet 3
CLOCK	Octet 4
PROTOCOL	Octet 5
INTERFACE	Octet 6
IDLE DATA (optional)	Octet 7
BUFFER SIZE (optional)	Octet 8

Figure 103/MTS22-C
Specify Synchronous Operating Mode command format

The SPEED octet specifies the interface baud rate, and shall be defined as follows:
Bits 8 to 5 = not used

b4	b3	b2	b1	= baud rate
0	0	0	0	no change
0	0	0	1	1200 bit/s
0	0	1	0	2400 bit/s
0	0	1	1	4800 bit/s
0	1	0	0	9600 bit/s
0	1	0	1	19200 bit/s
0	1	1	0	48 kbit/s
0	1	1	1	56 kbit/s
1	0	0	0	64 kbit/s

The CLOCK octet specifies the clock mode to be used, and is defined as follows:
Bits 8 to 4 = not used

b3	b2	b1	= clock mode
0	0	0	no change
0	0	1	internal (dataset source)
0	1	0	external (attached device source)
0	1	1	system (recovered)
1	0	0	dual external

The PROTOCOL octet specifies the B channel protocol to be used, and is defined as follows:
Bits 8,7 = not used

b6	b5	b4	b3	b2	b1	= B channel protocol
0	0	0	0	0	0	no change
0	0	0	0	0	1	MiNET synchronous
0	0	0	0	1	0	X.31

0	0	0	1	0	0	V.110
0	0	1	0	0	0	DMI mode 1
0	1	0	0	0	0	asynchronous or DS4113 mode
1	0	0	0	0	0	coax (MCX)

The INTERFACE octet specifies the interface type to be used, and is defined as follows:
Bits 8 to 4 = not used

b3	b2	b1	= interface type
0	0	0	no change
0	0	1	RS-232C
0	1	0	X.21
1	0	0	V.35

The optional IDLE DATA octet specifies the idle data pattern to be used by the terminal, and may take on any value in the range 0 to FF (hexadecimal). The default value shall be FF (HDLC idle).
The optional BUFFER SIZE octet specifies the buffer size for a terminal.
For the DS2200, the BUFFER SIZE octet may take on values in the range 2 to 4.
For the DS2100, the BUFFER SIZE octet represents the nominal receive buffer size at which output from the terminal begins. The phase locked loop speed adjustment thresholds are derived from this nominal value. If the value is set to 0 then the default value stored in ROM shall be used.
The size of the synchronous transport receive buffer is limited and baud rate dependent. Refer to PS.3 [14] for further details.

6.2.2.1.13 Request DS2100 Switch Settings (BF+14)

The format of this command shall be as shown in Figure 104/MTS22-C.
This command shall be used by the system to query the switch settings on a DS2100 data set. Optionally, the command may also be used to specify the packet size to be used in synchronous data transport. The terminal shall respond with a Report DS2100 Switch Settings response (see § 6.2.2.2.7).

MT	Octet 1
MT	Octet 2
PACKET SIZE (optional)	Octet 3

Figure 104/MTS22-C
Request DS2100 Switch Settings command format
The PACKET SIZE octet specifies the minimum packet size to be used for communication on a B channel, and may take on values in the range 0 to 10 (decimal). When the packet size parameter is equal to 0, the terminal shall use its default value (see PS.3 [14] for details).

6.2.2.1.14 Specify EIA Inputs Default ON (BF+15)

The format of this command shall be as shown in Figure 105/MTS22-C.
This command shall be used by the system to specify individual EIA inputs to be defaulted ON. The specified inputs shall always be reported as ON by the terminal, regardless of their actual state. This command differs from the Specify Interface Options command, in that the latter defaults all inputs ON.

MT	Octet 1
MT	Octet 2
EIA1	Octet 3
EIA2 (optional)	Octet 4

Figure 105/MTS22-C
Specify EIA Inputs Default ON command format

The EIA1 and optional EIA2 octets shall be encoded in accordance with § 6.2.1.2. The signals specified shall be defaulted ON. The EIA2 octet shall only be included if the extension bit in the EIA1 octet is set to 1.

6.2.2.1.15 Request TX/RX Character Count (BF+17)

The format of this message shall be as shown in Figure 106/MTS22-C.
This command shall be used by the system to query the transmit and receive character counts kept by a terminal. The characters counted are those received from and transmitted to the attached device.

MT	Octet 1
MT	Octet 2

Figure 106/MTS22-C
Request TX/RX Character Count command format

6.2.2.1.16 Specify EIA Outputs Default ON (BF+19)

The format of this command shall be as shown in Figure 107/MTS22-C.
This command shall be used by the system to specify individual EIA outputs to be defaulted ON.
During operation on the D channel, the system can specify EIA outputs ON or OFF via the Specify EIA Signals command (see § 6.2.2.1.5). However, once the terminal is operating on a B channel, the EIA outputs are controlled end to end by the terminals. The Specify EIA Outputs Default ON command allows the system to override the normal EIA signal
exchange.

MT	Octet 1
MT	Octet 2
EIA1	Octet 3
EIA2 (optional)	Octet 4

Figure 107/MTS22-C
Specify EIA Outputs Default ON command format
The EIA1 and optional EIA2 octets shall be encoded in accordance with § 6.2.1.2. The signals specified shall be defaulted ON. The EIA2 octet shall only be included if the extension bit in the EIA1 octet is set to 1.

6.2.2.1.17 Specify Disconnect Lead (BF+1A)

The format of this command shall be as shown in Figure 108/MTS22-C.
This command shall be used by the system to indicate to a terminal the interface leads to use as an indication to disconnect a B channel session.
This command's primary application is in the DS1100 and DS2100 data sets, which can disconnect on DTR or RTS. Both use DTR as the default in DCE mode, or DSR in DTE mode.

MT	Octet 1
MT	Octet 2
EIA octet 1	Octet 3

Figure 108/MTS22-C
Specify Disconnect Lead command format
The EIA1 octet shall be encoded as defined in § 6.2.1.2. If more than one signal is specified, then a disconnect shall occur when any of the signals drops.

Note - The EIA1 octet's extension bit shall have no meaning in this command.

On deactivation of the specified lead, the terminal shall disconnect from the B channel, re-connect to the D channel, and report the disconnect reason via the Report B To D Reason response (see § 6.2.2.2.4).

6.2.2.1.18 Specify RTS/CTS Delay (BF+1B)

The format of this command shall be as shown in Figure 109/MTS22-C.
This command shall be used by the system to specify the delay to activation of CTS following an activation of RTS. The default delay shall be 0 (immediate).
Note - The main application of this command is for multipoint operation of a DS2100 data set.

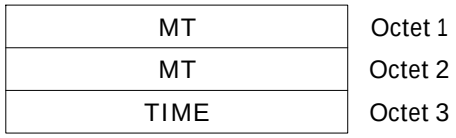


Figure 109/MTS22-C
Specify RTS/CTS Delay command format

The TIME octet specifies the time delay in milliseconds, and may take on values in the range 0 to 255 (decimal).

6.2.2.1.19 Request Packet Statistics (BF+1C)

The format of this command shall be as shown in Figure 110/MTS22-C.
This command shall be used by the system to collect statistics on packet and error counts related to the user port on DS2100 and DS2200 data sets. The terminal shall reply with a Report Packet Statistics response (see § 6.2.2.2.9).

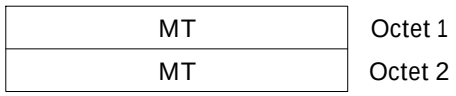


Figure 110/MTS22-C
Request Packet Statistics command format

6.2.2.1.20 Clear Network Management Statistics (BF+1E)

The format of this command shall be as shown in Figure 111/MTS22-C.
This command shall be used by the system to clear statistics counters maintained by a terminal.

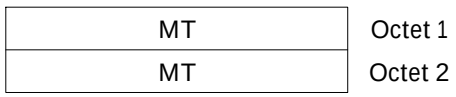


Figure 111/MTS22-C
Clear Network Management Statistics command format

This command clears statistics associated with the following terminal responses:
- Report Link Layer Statistics (see § 6.2.2.2.2);
- Report TX/RX Character Count (see § 6.2.2.2.8);
- Report Packet Statistics (see § 6.2.2.2.9); and
- Report Buffer Statistics (see § 6.2.2.2.10).

6.2.2.1.21 Specify DS2200 Data Mask (BF+1F)

The format of this command shall be as shown in Figure 112/MTS22-C.
This command shall be used by the system to specify a binary mask which the terminal will use to exclusive-OR all outgoing data. This command provides the capability to invert individual data bits

selectively.

MT	Octet 1
MT	Octet 2
MASK	Octet 3

Figure 112/MTS22-C
Specify DS2200 Data Mask command format

6.2.2.1.22 Request Buffer Statistics (BF+21)

The format of this command shall be as shown in Figure 113/MTS22-C.
This command shall be used by the system to collect statistics on usage of a terminal's internal buffers.

MT	Octet 1
MT	Octet 2

Figure 113/MTS22-C
Request Buffer Statistics command format

6.2.2.1.23 Specify Buffer / MiLAP Parameters (BF+22)

The format of this command shall be as shown in Figure 114/MTS22-C.
This command shall be used by the system to specify the following parameters to a terminal:

- thresholds at which flow control will be activated and deactivated to the attached device;
- thresholds at which data link layer flow control will be activated and deactivated; and
- data link layer operating parameters for the current session.

Note - This command is not supported by the DS2200 data set.

MT	Octet 1
MT	Octet 2
XOFF (optional)	Octet 3
XON (optional)	Octet 4
RNR ON (optional)	Octet 5
RNR OFF (optional)	Octet 6
MAX I FIELD (optional)	Octet 7
WINDOW (optional)	Octet 8
T200 (optional)	Octet 9
N200 (optional)	Octet 10

Figure 114/MTS22-C
Specify Buffer / MiLAP Parameters command format

The XOFF octet specifies the threshold at which a terminal will transmit an XOFF character or drop CTS to the attached device. The mode depends on the flow control mechanism specified in the Specify Interface Options command (see § 6.2.2.1.6).

The XON octet specifies the threshold at which a terminal will transmit an XON character or raise CTS to the attached device. The mode depends on the flow control mechanism specified in the Specify Interface Options command (see § 6.2.2.1.6).

The 6 octets described below are used to configure parameters associated with the data link layer. Refer to MTS21 [10] for further details.
The RNR ON octet specifies the number of I frames the terminal is to buffer before transmitting an RNR frame (i.e. enter local busy state) to the remote data link layer entity.
The RNR OFF octet specifies the number of buffered I frames at which the terminal will transmit an RR frame (i.e. exit local busy state) to the remote data link layer entity.
The MAX I FIELD octet specifies the maximum length in octets of an I (information) field to be transmitted by the terminal.

The WINDOW octet specifies the maximum number of outstanding (unacknowledged) I frames to be allowed by the terminal.
The T200 octet specifies the length of the T200 timer in steps of 10 ms (e.g. a decimal value of 10 indicates 100 ms).
The N200 octet specifies the number of times the terminal is to attempt to retransmit an unacknowledged I frame.
Each of the 8 parameters described above may take on values (decimal) as follows:

0 = no change
1 to 254 = valid values
255 = return to default value.

The default values (decimal) for each parameter shall be as shown below:

XOFF = 175
XON = 75
RNR ON = 13
RNR OFF = 9
MAX I FIELD = 13
WINDOW = 4
T200 = 50 ms or 300 ms (D channel or B channel with Go-ahead protocol)
N200 = 7

Note1 - For a description of the Go-ahead protocol, refer to MTS20 [9].

Note2 - The default values are restored each time a terminal makes a transition to the D channel.

6.2.2.1.24 Specify Hardware Flow Control (BF+23)

The format of this command shall be as shown in Figure 115/MTS22-C.
The use of this command is optional. When required, it shall be used by the system to further specify the method of hardware flow control if the CTS flow control option has been previously set using the Specify Interface Options command (see § 6.2.2.1.6).

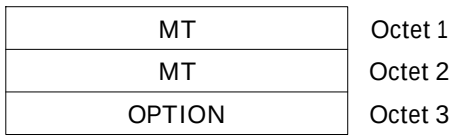


Figure 115/MTS22-C
Specify Hardware Flow Control command format
The OPTION octet specifies the hardware flow control as follows:
Bits 8 to 3 = reserved (must be set to 0)

b2	b1	= hardware flow control
0	0	no change
0	1	CTS only
1	0	EIA pin 25 HIGH / CTS
1	1	EIA pin 25 LOW / CTS

Note - The terminal's hardware flow control option will be reset to the default of CTS only whenever a

Specify Interface Options command (see § 6.2.2.1.6) with a flow control option other than no change is received from the system.

6.2.2.1.25 Specify D Channel Options (BF+24)

The format of this command shall be as shown in Figure 116/MTS22-C.
This command shall be used by the system to specify various D channel operating parameters for a terminal.

MT	Octet 1
MT	Octet 2
OPTION	Octet 3

Figure 116/MTS22-C

Specify D Channel Options command format

The OPTION octet specifies D channel options as follows:
Bits 8 to 5 = reserved (must be set to 0)

b4 b3 = flow control option
- 0 0 no change
- 0 1 enable flow control on the D channel
- 1 0 disable flow control on the D channel
- 1 1 reserved

b2 b1 = ready LED operation
- 0 0 no change
- 0 1 turn ready LED ON while operating on D channel
- 1 0 turn ready LED OFF while operating on D channel
- 1 1 reserved

Note1 - The D channel flow control method shall be as defined by the Specify Interface Options command (see § 6.2.2.1.6) and, if necessary, the Specify Hardware Flow Control command (see § 6.2.2.1.24).
Note2 - Whenever a transition occurs from the D channel to a B channel, the options will reset to their default values of flow control disable and ready LED OFF.

6.2.2.1.26 Specify DTR Disconnect Timer (BF+25)

The format of this message shall be as shown in Figure 117/MTS22-C.
This command shall be used by the system to specify the DTR disconnect timer value. When enabled, this timer provides a debounce function on the ON-to-OFF transition of the DTR lead, preventing spurious disconnects (see further explanation
below).

MT	Octet 1
MT	Octet 2
DTRDTV	Octet 3

Figure 117/MTS22-C

Specify DTR Disconnect Timer command format

The DTRDTV (DTR Disconnect Value) octet specifies the disconnect timer, and may take on values (decimal) as follows:
- 0 = timer disabled
- 1 to 255 = valid timer values in seconds

When the DTR signal has been selected via the Specify Disconnect Lead command (see § 6.2.2.1.17), an ON-to-OFF transition of the signal causes the terminal to disconnect from the B channel connection. However, in an electrically noisy environment, the DTR signal may exhibit short spurious transitions. The DTR disconnect timer causes the terminal to ignore signal transitions of a duration shorter than the timer value.

Note - The use of this timer does not conform to standard interface procedures, however, its operation does not affect compatibility.

6.2.2.1.27 Specify Interface Type (BF+26)

The format of this command shall be as shown in Figure 118/MTS22-C.
This command shall be used by the system to inform the terminal of the attached device's interface type.

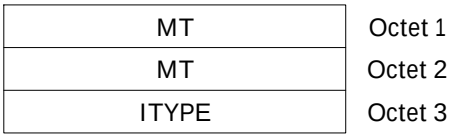


Figure 118/MTS22-C
Specify Interface Type command format

The ITYPE (Interface Type) octet defines the interface type according to the following:
Bits 8 to 5 = not used
Bits 4 to 1 = interface type

b4	b3	b2	b1	
0	0	0	0	no change
0	0	0	1	RS-232-C [21]
0	0	1	0	X.21 [4]
0	1	0	0	V.35 [3]
1	0	0	0	EIA-232-D (V.24 [2])

Note - The first 4 codings are compatible with the Specify Synchronous Operating Mode command (see § 6.2.2.1.12.5). Since both commands may be used to specify an interface type, the latest command issued will take precedence.

6.2.2.2 Responses from terminal to system

6.2.2.2.1 Report RS-232 Interface State (BF+0B)

The format of this command shall be as shown in Figure 119/MTS22-C.
This response shall be used by a terminal to reply to the system's Request RS-232 Interface State command (see § 6.2.2.1.1).

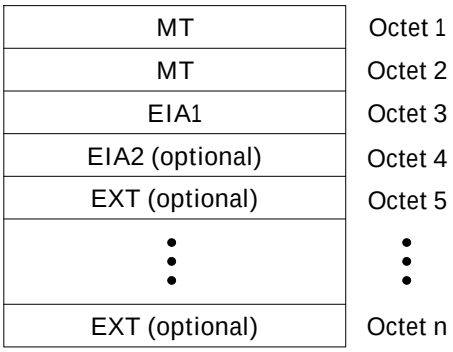


Figure 119/MTS22-C
Report RS-232 Interface State response format

The EIA1 and optional EIA2 octets shall be encoded in accordance with § 6.2.1.2. The EIA2 octet shall only be included when the extension bit in the EIA1 octet is set to 1.

Note1 - At this time, bits 7 to 1 of the EIA2 octet shall be set to 0. The extension bit (bit 8) of the EIA2 octet shall be used to indicate the presence or absence of an optional EXT octet.
Note2 - This command may also be used for interfaces other than RS-232 (e.g. V.35), however, the mapping of bits to interface signals shall not form part of this specification.

The EXT (Extension) octets allow for the transfer of additional information. Specific encodings are for further study.

Note - A terminal may also report its interface status via the RS-232 Status And ASCII Data message (see § 6.2.1.4).

6.2.2.2.2 Report Link Layer Statistics (BF+0C)

The format of this response shall be as shown in Figure 120/MTS22-C.
This response shall be used by a terminal in reply to a Request Link Layer Statistics command (see § 6.2.2.1.2).

MT	Octet 1
MT	Octet 2
RX FRAMES	Octet 3
⋮	⋮
RX FRAMES	Octet 6
CRC ERROR	Octet 7
CRC ERROR	Octet 8
LINK RESET	Octet 9

Figure 120/MTS22-C

Report Link Layer Statistics response format

The 4 RX FRAME octets report a count of data link layer frames received.
The 2 CRC ERROR octets report a count of data link layer CRC errors or frame aborts detected.
The LINK RESET octet reports a count of the data link layer resets detected.

Note1 - The counters shall stop at their maximum value to indicate an overflow condition.
Note2 - In general, the counters shall apply to frames received over both the D and B channels. For terminals which do not operate a data link layer on a B channel (DS2200 and some DS2100 modes), the counters shall apply to D channel frames only.
Note3 - The counters shall be cleared on loss of DNIC sync, loss of MiLINK physical layer (see MTS20 [9]), and on receipt of a Clear Network Management Statistics command (see § 6.2.2.1.20).
Note4 - For further information on operation of the data link layer, refer to MTS21 [10].

6.2.2.2.3 Report Baud Rate Information (BF+0D)

The format of this response shall be as shown in Figure 121/MTS22-C.
This command shall be used by a terminal to report its current baud rate.

MT	Octet 1
MT	Octet 2
BR	Octet 3

Figure 121/MTS22-C

Report Baud Rate Information response format

The BR (Baud Rate) octet indicates the current baud rate, and may take on values (hexadecimal) as shown below:

00 = no change
01 = 50 bit/s

02 = 75 bit/s
10 = 110 bit/s
11 = 134.5 bit/s
20 = 150 bit/s
30 = 200 bit/s
40 = 300 bit/s
50 = 600 bit/s
60 = 1200 bit/s
61 = 1800 bit/s
62 = 2000 bit/s
70 = 2400 bit/s
71 = 3600 bit/s
80 = 4800 bit/s
81 = 7200 bit/s
90 = 9600 bit/s
91 = 14400 bit/s
92 = 16600 bit/s
A0 = 19200 bit/s

6.2.2.2.4 Report B To D Reason (BF+0E)

The format of this response shall be as shown in Figure 122/MTS22-C.
This response shall be used by a terminal to indicate the reason for a disconnect from a B channel and re-establishment on the D
channel.

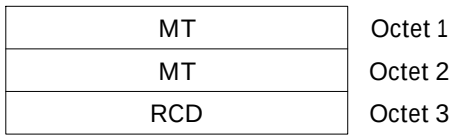


Figure 122/MTS22-C

Report B To D Reason response format

The RCD (Reason Code) octet specifies the disconnect reason, and may take on values (hexadecimal) as shown below:

00 = reserved
01 = data link layer failure
02 = data link layer abort
03 = data link layer DISC received
04 = control lead dropped from attached device
05 = terminal's CALL/ATTN key activated
06 = BREAK condition detected
07 = session activity timer expired
08 = attention character detected
09 = unsupported message received (see § 3.4 and § 4.2.2.2)
0A = terminal's DISC key activated
0B = B channel disconnect requested
0C = connect confirmation timer expired
0D = protocol mismatch with attached device
0E = incompatible peer terminals
0F to FF = not used

Note1 - Reason 01 applies to situations such as receiving idle data link layer when flags are expected.
Note2 - Reason 02 applies to situations which involve data link layer errors such as failing to achieve data link layer reset after N200 attempts.
Note3 - For further information on operation of the data link layer, refer to MTS21 [10].

6.2.2.2.5 Report Keystroke (BF+0F)

The format of this response shall be as shown in Figure 123/MTS22-C.
This response shall be used by a terminal to inform the system of a depression of either the ATTN (Attention) or DISC (Disconnect) key.

Note - This response pertains only to the stand-alone and cartridge versions of data sets.

MT	Octet 1
MT	Octet 2
KEY	Octet 3

Figure 123/MTS22-C

Report Keystroke response format
The KEY octet specifies the key which has been depressed. Depression of a key shall be indicated by setting a bit to 1 as indicated below:

Bit 8 = DISC
Bit 7 = CALL/ATTN
Bits 6 to 1 = not used

6.2.2.2.6 Report DS2200 Network Sync Timing (BF+13)

The format of this response shall be as shown in Figure 124/MTS22-C.
This response shall be used by a terminal in reply to a Specify DS2200 Network Sync Timing command from the system (see § 6.2.2.1.12).

MT	Octet 1
MT	Octet 2
NETSYNC MT	Octet 3
PARAMETER (optional)	Octet 4
⋮	⋮
PARAMETER (optional)	Octet n

Figure 124/MTS22-C

Report DS2200 Network Sync Timing response format
This response is actually composed of 7 sub-responses, determined by the value of the NETSYNC MT octet as follows:

NETSYNC MT	Sub-response
1	Acknowledge Network Synchronization Disable
2	Acknowledge Network Synchronization Enable
3	Acknowledge Request Long Term Accumulator
4	Report Clock Source
5	Report Short Term Accumulator
6	Report Long Term Accumulator
7	Request Synchronization

The sub-responses are defined in §§ 6.2.2.2.6.1 to 6.2.2.2.6.7.
The optional PARAMETER octets have specific meanings as defined in the individual sub-responses.

6.2.2.2.6.1 Acknowledge Network Synchronization Disable

NETSYNC MT = 1
This sub-response shall be used by a terminal in reply to a Disable Network Synchronization sub-command from the system (see § 6.2.2.1.12.1).

Note - This sub-response does not include parameters.

6.2.2.2.6.2 Acknowledge Network Synchronization Enable

NETSYNC MT = 2
This sub-response shall be used by a terminal in reply to an Enable Network Synchronization sub-command from the system (see § 6.2.2.1.12.2).

Note - This sub-response does not include parameters.

6.2.2.2.6.3 Acknowledge Request Long Term Accumulator

NETSYNC MT = 3
This sub-response shall be used by a terminal in reply to a Request Long Term Accumulator command from the system (see § 6.2.2.1.12.3).

Note - This sub-response does not include parameters.

6.2.2.2.6.4 Report Clock Source

NETSYNC MT = 4
The format of this sub-response shall be as shown in Figure 125/MTS22-C.
This sub-response shall be used by a terminal in reply to a Request Clock Source sub-command from the system (see § 6.2.2.1.12.4).

MT	Octet 1
MT	Octet 2
NETSYNC MT	Octet 3
NET CLOCK STATE	Octet 4

Figure 125/MTS22-C

Report Clock Source sub-response format

The terminal shall signal its ability to act as a synchronization source if all of the following conditions are met:

- the terminal is in network timing mode;
- the network clock is present from the interface; or
- the interface is in a ready state.

If all the conditions are met, the terminal shall return the NET CLOCK STATE octet with a value of 1. If any of the conditions are not met, the terminal shall return the NET CLOCK STATE octet with a value of 0.

6.2.2.2.6.5 Report Short Term Accumulator

NETSYNC MT = 5
The format of this sub-response shall be as shown in Figure 126/MTS22-C.
This sub-response shall be sent by a terminal every 12.8 seconds while its network synchronization process is in the Acquisition state.

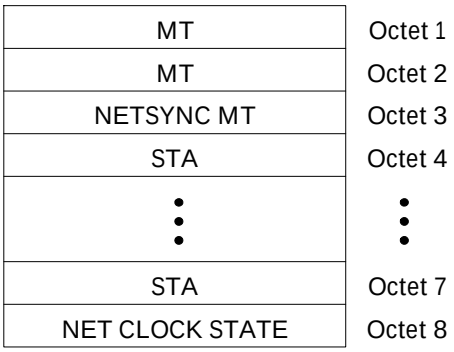


Figure 126/MTS22-C
Report Short Term Accumulator sub-response format
The 4 STA (Short Term Accumulator) octets contain the latest value of the terminal's short term accumulator.
The NET CLOCK STATE octet indicates the acceptability of the clock source, and shall be encoded as defined in § 6.2.2.2.6.4.

6.2.2.2.6.6 Report Long Term Accumulator

NETSYNC MT = 6
The format of this sub-response shall be as shown in Figure 127/MTS22-C.
This sub-response shall be sent by a terminal every 12.8 seconds while its network synchronization process is in the Tracking state.

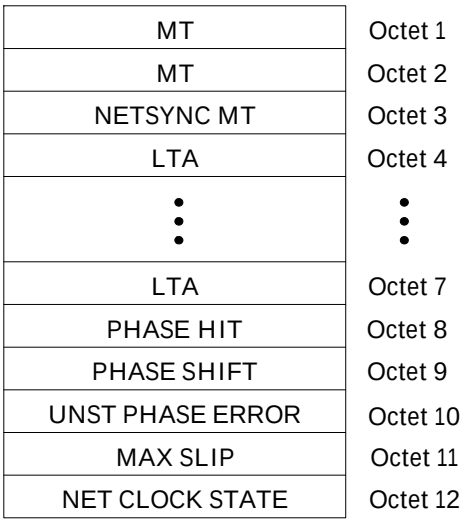


Figure 127/MTS22-C
Report Long Term Accumulator sub-response format
The 4 LTA (Long Term Accumulator) octets contain the latest value of the terminal's long term accumulator.
The PHASE HIT octet contains a cumulative count of phase hits.
The PHASE SHIFT octet contains a cumulative count of phase shifts.
The UNST (Unstable) PHASE ERROR octet contains a cumulative count of unstable phase errors.

Note - The above three counters shall roll over to zero and continue to count after they have reached maximum count.

The MAX SLIP octet indicates whether the maximum slip threshold has been exceeded. If the threshold has been exceeded, the value of the MAX SLIP octet is set to 1, otherwise the value is set to 0.
The NET CLOCK STATE octet indicates the acceptability of the clock source, and shall be encoded as defined in § 6.2.2.2.6.4.

6.2.2.2.6.7 Request Synchronization
NETSYNC MT = 7
This sub-response shall be sent by a terminal every 12.8 seconds while its network synchronization process is in the No-synchronization state. The terminal enters this state after reset, and shall send this sub-response continuously until it receives either a Disable Network Synchronization sub-command (see § 6.2.2.1.12.1) or an Enable Network Synchronization sub-command (see § 6.2.2.1.12.2).

Note - This sub-response does not include parameters.

6.2.2.2.7 Report DS2100 Switch Settings (BF+16)
The format of this response shall be as shown in Figure 128/MTS22-C.
This response shall be used by a DS2100 data set to report user-defined switch settings.

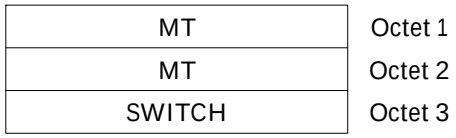


Figure 128/MTS22-C
Report DS2100 Switch Settings response format
The SWITCH octet indicates the current switch settings, and is defined as follows:
b8 b7 = operating mode
- x 0 asynchronous
- x 1 synchronous
- 0 x terminal to system
- 1 x terminal to terminal

Note - The terminal to terminal mode indicates the DS2100 is set to operate in a point-to-point configuration without the system. However, if the terminal finds DNIC synchronization as a slave, then it will operate in a terminal to system mode.

b6 b5 = clock source
- 0 0 internal clock
- 0 1 external clock
- 1 0 system clock
- 1 1 TX and RX external clock

b4 = B channel mode
- 0 MiNET synchronous
- 1 X.31

b3 b2 b1 = speed
- 0 0 0 1200 bit/s
- 0 0 1 2400 bit/s
- 0 1 0 4800 bit/s
- 0 1 1 9600 bit/s
- 1 x x 19200 bit/s

Note - The definitions of bits 6 to 1 are valid only if bit 7 indicates the synchronous operating mode.

6.2.2.2.8 Report TX/RX Character Count (BF+18)

The format of this response shall be as shown in Figure 129/MTS22-C.
This response shall be used by a terminal in reply to a Request TX/RX Character Count command from the system (see § 6.2.2.1.15).

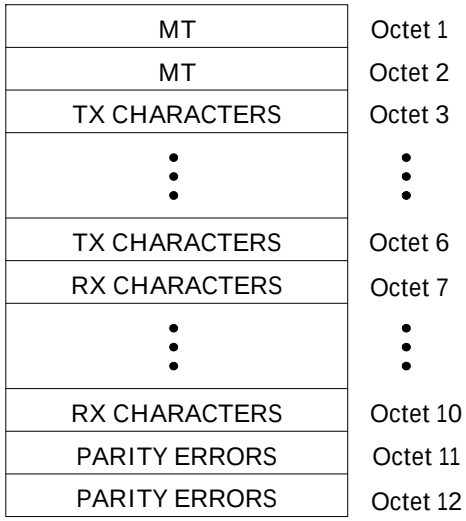


Figure 129/MTS22-C

Report TX/RX Character Count response format

The 4 TX CHARACTERS octets indicate the number of characters sent to the terminal from the attached device.

The 4 RX CHARACTERS octets indicate the number of characters sent to the attached device by the terminal.

The 2 PARITY ERRORS octets indicate the number of parity errors detected by the terminal from the attached device.

Note1 - The PARITY ERRORS octets are only valid if the non-transparent parity mode has been set via a Specify UART Parameters command (see § 6.2.2.1.7).
Note2 - The above counters shall stop at their highest count to indicate an overflow condition.
Note3 - The counters shall be cleared on loss of DNIC sync, loss of MiLINK physical layer (see MTS20 [9]), and on receipt of a Clear Network Management Statistics command (see § 6.2.2.1.20).

6.2.2.2.9 Report Packet Statistics (BF+1D)

The format of this response shall be as shown in Figure 130/MTS22-C.
This command shall be used by a terminal to report packet statistics in reply to a Request Packet Statistics command (see § 6.2.2.1.19). This response is applicable when the terminal is operating with a packet-mode attached

device.

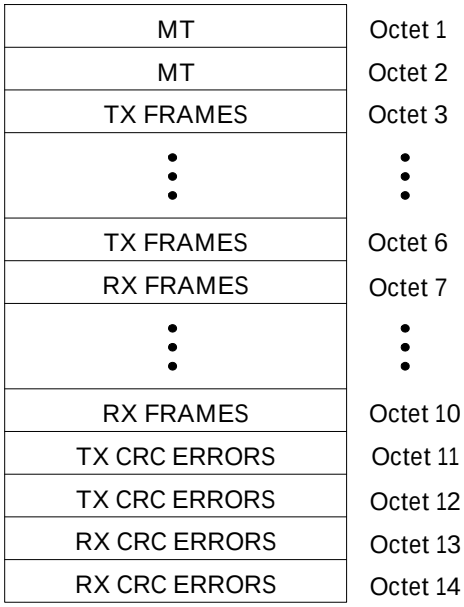


Figure 130/MTS22-C

Report Packet Statistics response format

The 4 TX FRAMES octets indicate the number of frames sent to the terminal from the attached device.
The 4 RX FRAMES octets indicate the number of frames sent to the attached device from the terminal.
The 2 TX CRC ERRORS octets indicate the number of frames received with CRC errors or frame aborts by the terminal from the attached device.
The 2 RX CRC ERRORS octets indicate the number of frames received with CRC errors or frame aborts by the terminal from the system.

Note1 - The above counters shall stop at their highest count to indicate an overflow condition.
Note2 - The counters shall be cleared on loss of DNIC sync, loss of MiLINK physical layer (see MTS20 [9]), and on receipt of a Clear Network Management Statistics command (see § 6.2.2.1.20).

6.2.2.2.10 Report Buffer Statistics (BF+20)

The format of this response shall be as shown in Figure 131/MTS22-C.
This response shall be used by a terminal in response to a Request Buffer Statistics command from the system (see § 6.2.2.1.22).

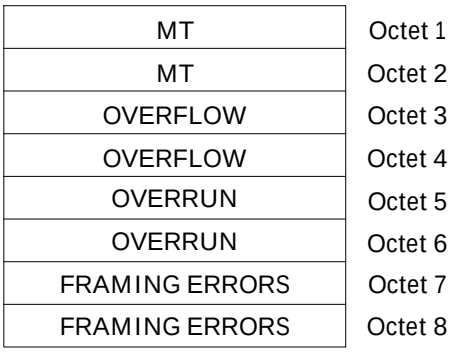


Figure 131/MTS22-C

Report Buffer Statistics response format
The 2 OVERFLOW octets indicate buffer overflows. The meaning is dependent on the terminal type, and is defined as follows:

DS2200 = number of slips
DS2100 (sync. mode) = number of PLL buffer overflows
DS2100 (async. mode) = number of receive information overflows (from attached device)
DS1100 = number of receive information overflows (from attached device)

The 2 OVERRUN octets indicate the number of UART overrun errors on data received from the attached device. The counts apply only to the DS2100 (async mode) and the DS1100.
The two FRAMING ERRORS octets indicate the number of UART framing errors on data received from the attached device. The counts apply to the DS2100 (async mode) and the DS1100.

6.2.3 MCX messages

The commands and responses described in this section are used between the DS4122 controller data set and the DS4113 terminal data set. The MCX protocol allows for coaxial cable elimination between IBM 3274 cluster controllers and its attached display devices. Refer to PS.4 [15] for further details.

6.2.3.1 Messages from controller to terminal

6.2.3.1.1 Command Frame (82)

6.2.3.1.1.1 Definition

The format of this message shall be as shown in Figure 132/MTS22-C.
This message shall be used by the controller data set to send a series of IBM commands to the terminal data set.

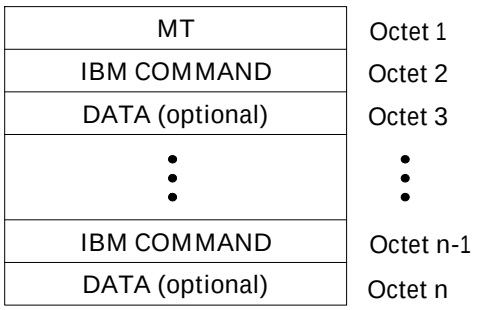


Figure 132/MTS22-C
Command Frame message format
The value of n shall not exceed 14 (decimal).
Bits 8 to 1 of each IBM COMMAND octet represent bits 2 to 9 of an IBM command. Each IBM COMMAND octet may be followed by an optional DATA octet.

6.2.3.1.1.2 Interpretation octets

The MCX Command Frame takes advantage of the fact that the action required for any given IBM command is fixed. The IBM COMMAND octets in the command are not only sent from the terminal (DS4113) to the attached device, but are also used by the terminal as an index to an interpretation table. Each IBM COMMAND octet has a corresponding octet stored in the interpretation table. The interpretation table is downloaded from the controller data set (DS4122) to the terminal data set (DS4113) at call setup. The interpretation octets shall be encoded as shown below:
b8 = P12

0 IBM bit 12 set to test mode
1 IBM bit 12 set to normal mode (bit 12 is parity)

b7 = R

0 controller data set does not expect a response
1 controller data set requires a response

Note - The required response shall be a Solicited Response (see § 6.2.3.2.1).

b6 = B

0 command will not cause a busy condition in the attached device
1 command will cause a busy condition in the attached device

Note - If b6 is set to 1 and the attached device does not return an Operation Complete response within 50 ms, the terminal data set shall send an Error response (see § 6.2.3.2.3).

b5 = p10

0 odd parity shall be sent in IBM bit 10
1 IBM bit 10 transparent; D10 shall be sent in IBM bit 10 (see below)

b4 = NRe

0 terminal data set may retry if command fails
1 terminal shall not retry command after failure

b3 = D

0 following octet is command
1 following octet is data

Bit 2 = D10

This bit contains the value of the IBM bit 10, which shall be sent to the attached device if p10 is set to 1.

Bit 1 = D11

This bit contains the value of the IBM bit 11, which shall be sent to the attached device.

6.2.3.1.2 Data Frame (83)

The format of this message shall be as shown in Figure 133/MTS22-C.

This message shall be used by the terminal to send data to the attached device for display. The terminal shall retransmit a failing sequence up to the maximum number of retries prior to sending an Error response (see § 6.2.3.2.3) to the controller data set.

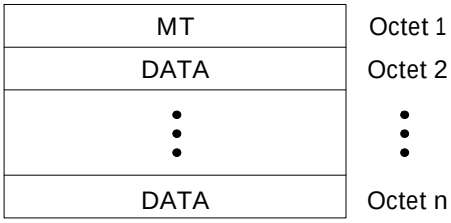


Figure 133/MTS22-C

Data Frame message format

The value of n shall not exceed 14 (decimal).

For attached devices that do not support an Extended Attribute Buffer (EAB), bits 8 to 1 of each DATA octet shall contain bits 2 to 9 of an IBM data word.

For attached devices that support an EAB, the DATA octets shall be paired. Bits 8 to 1 of the first DATA octet shall contain bits 2 to 9 of the IBM data word to be written to screen RAM. Bits 8 to 1 of the second DATA octet shall contain bits 2 to 9 of the IBM data word to be written to EAB RAM.

6.2.3.1.3 General Frame (84)

The general format of this message shall be as shown in Figure 134/MTS22-C.

This message shall be used by the controller data set to control various aspects of the terminal data set or attached

device.

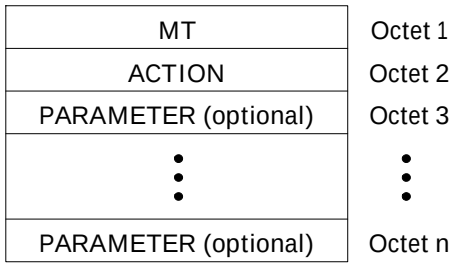


Figure 134/MTS22-C

General Frame message format

The value of n shall not exceed 14 (decimal).

The ACTION octet specifies the actions to be performed, and may take on values (hexadecimal) as shown below:

01 = Literal Command Sequence
02 = Turn Poll ON
03 = Turn Poll OFF
04 = Retry Number
05 = Download Command Interpreter
06 = Direct Data To Buffer
07 = Transfer Buffer To Display
08 = Time To Wait For Operation Completion
09 = Clear Command Interpreter Table
0A = Write Spaces

The actions are defined in §§ 6.2.3.1.3.1 to 6.2.3.1.3.10.

6.2.3.1.3.1 Literal Command Sequence

ACTION code = 01

The format of this message shall be as shown in Figure 135/MTS22-C.

This message shall be used to send literal commands.

Note - This message is not used at the current time. The Command Frame message (see § 6.2.3.1.1) is preferred.

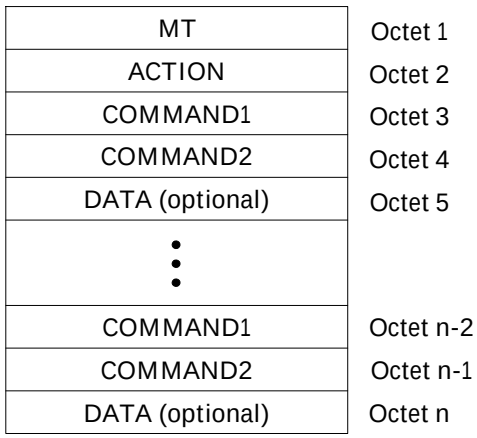


Figure 135/MTS22-C

Literal Command Sequence message format

The COMMAND1 octet specifies the command interpretation, and shall be encoded as defined in § 6.2.3.1.1.2.

Bits 8 to 1 of the COMMAND2 octet contain bits 2 to 9 of the IBM command.

Each COMMAND2 octet may be followed by an optional DATA octet. Bits 8 to 1 of the DATA octet shall contain bits 2 to 9 of the IBM data.

6.2.3.1.3.2 **Turn Poll ON**
ACTION code = 02
This message shall be used by the controller data set to cause the terminal data set to begin polling the attached device. All non-zero poll responses from the attached device shall be sent by the terminal to the controller data set via the Unsolicited Response (see § 6.2.3.2.2).
This message does not contain parameters.

6.2.3.1.3.3 **Turn Poll OFF**
ACTION code = 03
This message shall be used by the controller data set to cause the terminal data set to cease polling the attached device.
This message does not contain parameters.

6.2.3.1.3.4 **Retry Number**
ACTION code = 04
The format of this message shall be as shown in Figure 136/MTS22-C.
This message shall be used by the controller data set to specify to the terminal data set the number of retries to be attempted to the attached device prior to sending an Error response.

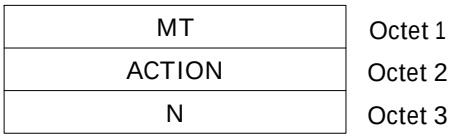


Figure 136/MTS22-C
Retry Number message format
The N octet specifies the number of retries and may take on values (decimal) as follows:
0 = terminal shall transmit every response from the attached device to the controller data set
1 to 255 = valid retry numbers

Note - This message is not currently supported in the DS4113.

6.2.3.1.3.5 Download Command Interpreter
ACTION code = 05
The format of this message shall be as shown in Figure 137/MTS22-C.
This message shall be used by the controller data set to download the command interpreter tables to the terminal.

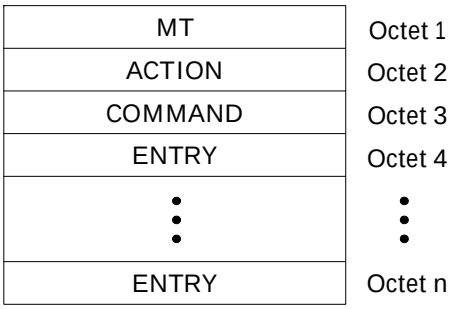


Figure 137/MTS22-C
Download Command Interpreter message format
The maximum value of n shall be 14 (decimal).

Bits 8 to 1 of the COMMAND octet contain bits 2 to 9 of the IBM command. This value shall constitute the start address of the interpretation table for the first download entry. Subsequent entries are added to the table in order.
The ENTRY octets shall be encoded as defined in § 6.2.3.1.1.2, and shall be stored in the interpretation table starting at the address specified by the COMMAND octet.

6.2.3.1.3.6 Direct Data To Buffer

ACTION code = 06
The format of this message shall be as shown in Figure 138/MTS22-C.
This message shall be used by the controller data set to write data into a terminal data set's RAM buffer, rather than the attached device.

Note - This message is not currently supported on either the DS4122 or the DS4113.

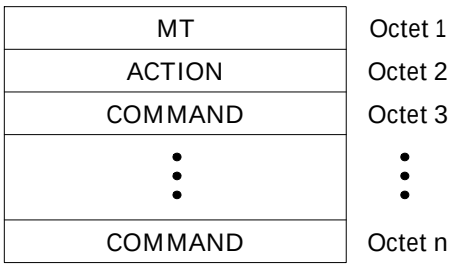


Figure 138/MTS22-C
Direct Data To Buffer message format

6.2.3.1.3.7 Transfer Buffer To Display

ACTION code = 07
This message shall be used by the controller data set to instruct the terminal data set to transfer the contents of its RAM buffer to the attached device.
This message does not contain parameters.

Note - This command is not currently supported on either the DS4122 or the DS4113.

6.2.3.1.3.8 Time To Wait For Operation Completion

ACTION code = 08
This message shall be used by the controller data set to specify the length of time to allow while waiting for a “busy” type of operation (e.g. clear, insert, search, etc.) to
complete.

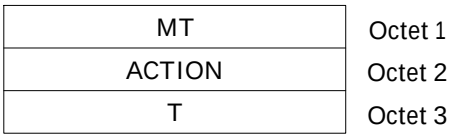


Figure 139/MTS22-C
Time To Wait For Operation Completion message format
The T octet specifies the wait time in ms, and may take on values in the range 0 to 255 (decimal).

Note - This message is currently not supported in the DS4113.

6.2.3.1.3.9 Clear Command Interpreter Table

ACTION code = 09

This message shall be used by the controller data set to instruct the terminal data set to clear its command interpreter table.
This message does not contain parameters.

Note - This message is not currently supported by the DS4113.

6.2.3.1.3.10 Write Spaces

ACTION code = 0A
The format of this message shall be as shown in Figure 140/MTS22-C.
This message shall be used by the controller data set to instruct the terminal data set to write a number of spaces to the attached device, starting at the current cursor
position.

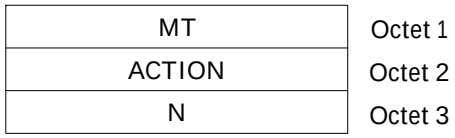


Figure 140/MTS22-C
Write Spaces message format
The N octet specifies the number of spaces, and may take on values in the range 0 to 255 (decimal).

6.2.3.1.4 Write Extended Attribute Buffer Null (88)

The format of this message shall be as shown in
When an attached device supports an Extended Attribute Buffer (EAB), this command shall be used by the controller data set to instruct the terminal data set to write null values into specified EAB locations.

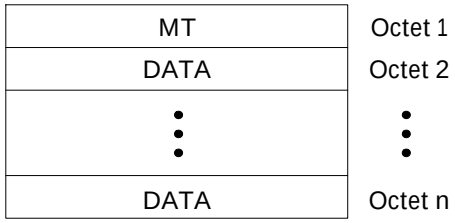


Figure 141/MTS22-C
Write Extended Attribute Buffer Null message format
The maximum value of n shall be 14 (decimal).
Bits 8 to 1 of the DATA octets contain bits 2 to 9 of IBM data (to be written to screen RAM).

6.2.3.2 Messages from terminal to controller

6.2.3.2.1 Solicited Response (82)

The format of this message shall be as shown in Figure 142/MTS22-C.
This message shall be used by the terminal data set to reply to commands (e.g. read) from the controller data set. The triggering command is returned within the

response.

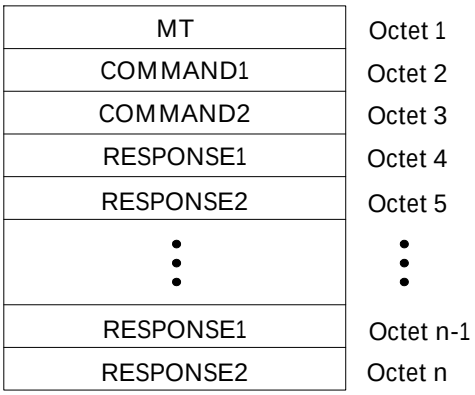


Figure 142/MTS22-C
Solicited Response message format
The maximum value of n shall be 11 (decimal). This allows for up to 4 RESPONSE octet pairs.
The COMMAND1 and COMMAND2 octets shall be encoded as defined in § 6.2.3.1.3.1.

The RESPONSE1 octet shall be encoded as follows:
Bits 8,7,5,2,1 = reserved
b6 = ERR (Error)
0 no error
1 error detected in response
Bit 4 = IBM data bit 10
Bit 3 = IBM data bit 11
Bits 8 to 1 of the RESPONSE2 octet contain IBM data bits 2 to 9.

6.2.3.2.2 Unsolicited Response (83)
The format of this message shall be as shown in Figure 143/MTS22-C.
This message shall be sent to the controller data set by the terminal data set each time a non-zero poll response is received from the attached device. Typically, this information will be keystrokes, although abnormal error conditions may be received as well. The terminal data set shall automatically acknowledge receipt of the non-zero poll
response.

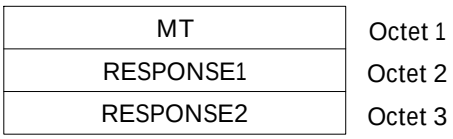


Figure 143/MTS22-C
Unsolicited Response message format
The RESPONSE1 and RESPONSE2 octets shall be encoded as defined in § 6.2.3.2.1.

6.2.3.2.3Error (84)
The format of this message shall be as shown in Figure 144/MTS22-C.
This message shall be used by the terminal data set to report errors to the controller data set.

MT	Octet 1
COMMAND1	Octet 2
COMMAND2	Octet 3
ERROR	Octet 4

Figure 144/MTS22-C
Error message format
The COMMAND1 and COMMAND2 octets shall be encoded as defined in § 6.2.3.1.3.1.
The ERROR octet defines the type of error detected, and shall be encoded as follows:
Bits 8 to 5 = not used

b4	b3	b2	b1	= error
x	x	x	1	maximum number of retries exceeded
x	x	1	x	response word bit violation error
x	1	x	x	time-out waiting for attached device response
1	x	x	x	time-out waiting for operation complete

Note - The controller data set shall commence recovery procedures on receipt of an error message. This shall normally consist of a reset of the attached device and retransmission of the current screen. In the event repetitive error messages are received, the controller data set shall terminate the B channel session, return to the D channel, and report the session termination reason to the system via the Report B To D Reason response (see § 6.2.2.2.4).

MiNET implementation guidelines

1.0 Overview

This annex describes the behaviour of voice sets using the MiNET protocol. It describes the interaction, interpretation or particular implementations of the sets.
Voice set operation must follow this annex where possible.

This annex was created as a result of efforts to design the 400 series of sets and simulated sets on workstations. Many operational details in the SUPERSETS 3DN and 4DN had never been documented. This annex reflects the current implementation of the 400 sets and the operation may depart from the SUPERSET 3DN and 4DN operation.

1.1 Glossary

Ringer combination
The combination of ringer tones and the alternation between these tones used as the ringing sound in the set. Also known as ringer pitch.

Undetermined behaviour
Means that the behaviour of the set in a particular circumstance is not deterministic and so must not be relied upon. It is up to the entity communicating with the set to ensure that these situations do not arise.

Product Specification
Response to MiNET Commands
This section describes the way the voice set behaves in response to MiNET Voice commands.

General

The set must support a maximum string length which is at least 2 characters longer than its screen width, in order to accommodate at least one flash sequence embedded in the string.

Clear screen
The command deactivates all prompts on the screen (if the set understands such a thing as a prompt).

Clear lines
When this is used on the lines where prompts are kept track of by the set, it implicitly deactivates all prompts on the affected lines.

Shift and Roll
Shift and roll commands with the ALL BUFFERS BCCB buffer code will be ignored by the set.

Send string
The top left of the screen is coordinate 1,1.
Strings can wrap around the end of one line to the start of the next line. A string can thus be longer than the actual display width. There is no benefit in this - it is just easier to implement in the set.
The effect of out of range row and column values is undetermined.

Hardware Control Byte (HCB)
Handset
The status of the hook-switch has no effect on the handset transducers. The set is to operate the two independently.

DTMF (line and local)
Enabling/disabling DTMF tones takes effect for the next keypad digit pressed. It does not affect a keypad depression in progress.
The set will produce a DTMF tone on the down-stroke of a keypad key if DTMF tones were previously enabled. The tone will last as long as the key is depressed and will be heard on the current output transducer (if local DTMF is enabled) - if the speaker is on, the tone will be sent to the speaker and if the earpiece then to the earpiece. The tone will also be sent down the B-channel at the same time (if line DTMF is enabled). If there are no output transducers enabled, the set will by default send the tone to the speaker. If the ringer is on then no DTMF tone is produced.
The gain for the DTMF tone will be relative to the current hands-free speaker setting, regardless of what the current audio mode is when the set produces the DTMF tone.
When the set produces the DTMF tones, the output transducer is “borrowed” for this. When the DTMF tone is done, the output transducer used must be restored to whatever its present MiNET state is (this includes the volume). The MiNET state is the status of the speaker or earpiece based upon the control of the MiNET commands. For example, if the speaker is turned off while the DTMF tone is being produced, the tone continues to completion and then the speaker is turned off.
If a ringer command (Start voice set or HCB ringer on) is received while producing a DTMF tone, the tone will be cancelled and the ringing heard as usual.

Ringer control
Turning on the ringer via the HCB command causes the ringer tone to be produced continuously. The set must not allow any breaks in the ringing tone, since the system may use ringer-on/ringer-off to cadence the ringing (instead of using Start voice set ringer). Ringer volume applies as with Start voice set ringer.
The ringer tone is produced on command and sent to the speaker. The ringer may run while the speaker is turned on. In this case, the speaker must be borrowed, as with the DTMF tone generation above. The volumes must be restored as above, so that when the ringer tone is produced, the ringer gain must be used and when it ends, the gain previously in use (the set may have had the speaker on and be in music audio mode) must be restored if the speaker is on. During the pause between ringing in the ring pattern, silence is heard (however the set achieves this) regardless of what the MiNET status of the speaker is (i.e. we don't want background music chopped up with a ring cadence).
Ringer commands received when the hands-free microphone is turned on will produce undetermined behaviour.

Ringer pattern commands can supersede each other, replacing the pattern currently in use. All timers involved are restarted.
A repeat period < 8 (1.6 s.) will produce undetermined behaviour.

Send DTMF
The set will produce a DTMF tone for 100ms. The tones specified in the command are separated by 100ms of silence. A 0FH produces 100ms of silence. The Acknowledge DTMF MiNET command is sent by the set at the end of the silence period.
Codes 0AF, 0BF, 0CF and 0DF are undetermined.
If a keypad digit is pressed while the set is producing a DTMF tone from this command, the command is aborted and the keypad key processed as usual. No acknowledgement is sent to the system regarding the Send DTMF command. The system can interpret the reception of a keypad event instead of the acknowledge as an abort of the tone production by the set.
The set will not queue up Send DTMF commands. The sender must wait for the set to send an Acknowledge DTMF command before sending another Send DTMF command. The behaviour of sending overlapping Send DTMF commands is undetermined.

Define Special character
Documents for each display set will specify how it supports definable characters. It will also specify whether the bytes for the command define rows or columns.

Audio modes
Ringer combination adjust
In this mode, the set is to ring with a Start voice set ringer combination of period=8 and cadence pattern=76H.
When the mode is entered, the status of ringing on the set is ignored. When the mode is exited, the ringer is turned off regardless of the MiNET status of the ringer and, as a side effect, the handset, microphone and speaker are all turned off and must be refreshed by the system.
A Start voice set ringer command received while in the ringer combination adjust mode will simply replace the ringer pattern currently being used and otherwise have no effect on the state of the set.
Use of the hands-free microphone in this mode will produce undetermined behaviour.
HCB ringer off in this mode is equivalent to a nil audio mode command.

Buffers and cursors
In current MiNET voice sets, the buffer and cursor numbers used in a BCCB must always match.

Flashing characters
When a flash attribute is specified for a character, the attribute stays with the character itself, and not the position on the display. For example, a flashing character can be shifted left and still remain flashing in its new position.

MiLINK and set reset
When a set resets, it must leave the MiLINK alone so that the devices on the link are not disturbed.

B channel handling
A voice set always defaults to the B1 channel.
Because of dynamic B channel allocation, the set must NOT use its B channel until the system explicitly turns on a transducer. The system may want to assign the set to the B2 channel after receiving the set's SETID.
If a set is denied a B channel, the set does not change its state. It is up to the system to flash LCDs, or put something on the set display. If the system turns on an audio transducer, the set does this regardless of whether or not a B channel has been assigned.
The behaviour of the set is unaffected by which B channel has been assigned to the set.

Initial configurations
Power-up
The set will have the following configuration on power-up, reset and requested reset:

DTMF tones on for non-consoles, off for consoles
all output transducers off
all delay updates cleared
buffer #1 will be mapped to the screen
cursor locations will be at 1,1
cursors off
the B1 channel is to be used

The following will be in an unknown state (the set may be using some of them as an indication of power-up etc.):

line LCDs
LEDs
LCD display
buffer contents
volumes and ringer combo

A qualification for the reset request is the handset volume. If it is being remembered by the set (it can do this and it has received a MiNET request to not reset to nominal all the time). The system cannot refresh this value and so the set must make every effort to save this value.

Loss of synchronisation

Everything should be reset to the power-up defaults as defined above except the handset volume if it is being remembered by the set (it can do this and it has received a MiNET request to not reset to nominal all the time). The system cannot refresh this value and so the set must make every effort to save this value.
Any D or B channels held must be released.
During loss of sync, the output transducers should be temporarily muted if any are enabled, to prevent hearing the possible DNIC noise that is generated during recovery.

Reporting a SETID

A SETID is to be reported:

after power-up. The reason code is hardware reset/ power-up.
after an internal error. The reason code is Software Trap.
after a loss of sync. The reason code is Loss of Sync.
after request. The reason code is #4.
after system reset. The reason code is #3.

Incorrect MiNET commands

The set is to reject all invalid MT codes received. The behaviour in response to incorrectly formatted contents of the MiNET commands is undetermined.

Decision alternatives

Minimum DTMF tone time

It was suggested that there be a minimum time for DTMF tones as a result of a keypad depression. This could produce a more reliable end-to-end signalling capability, and with a minimum time of 25-50 ms, it may not be that obvious to the end user.
It was decided that this would require a fair amount of user testing which we are unable to do at this time.

Down-strokes for the keypad

Only up-strokes are reported for dialled keys.

Appendix I- Example of precise ringer cadence setting

Appendix I provides an example of how to proceed to define the ringer cadencing for sets.

The example considered is the AT & T Call Back ringing pattern, defined as follows:

on	**off**	**on**	**off**	**on**	**off**	
0.4	0.2	0.4	0.2	0.8	4.0	Time intervals in seconds (Total time = 6 s)

As described in § 5.2.1.34, the ringer ON time interval shall therefore be set to 8 (400 ms) and the ringer OFF time interval shall be set to 4 (200 ms). The CPP would therefore be 84 in hexadecimal. The CPB corresponding to this ringing pattern will therefore be 10101100 (AC in hexadecimal), as shown in Table 14/MTS22-C:

Table 14/MTS22-C
Determination of CPB from desired ringer cadence

8	7	6	5	4	3	2	1	
0.4	0.2	0.4	0.2	0.4	0.4	0.2	0.2	Total time = 2.4 s
ON	OFF	ON	OFF	ON		OFF		ON/OFF sequence
1	0	1	0	1	1	0	0	CPB

There will be a 3.6 s (6.0 s - 2.4 s) missing (18 $\times$ 200 ms = 3.6 s) at the end of the pattern.
Using the formula in § 5.2.1.36, we

<Math?>

get:

<Math?>

In this case, the period is exactly matched since the RP is an integer (26 = 1A in hexadecimal).
The Start Precise Ringer message shall thus be coded as follows: 23 AC 1A 84 (in hexadecimal).
Peripherals that support the Start Precise Ringer message should also support the Start Voice Set Ringer message. Each CPB octet bit shall represent a 200 ms time period.

Section D - Conformance Testing

1. 0 Introduction

Please note that the tests in this section have not been updated to reflect version B04 of this protocol specification

1.1 Purpose

This document defines the abstract conformance test suite for MiNET, user or network side for messages pertaining to voice sets. These tests do not encompass the validation of B-channel functionality. The main purpose of these tests are to ensure that the MiNET messages conform to the MTS22-C specification. For example, string commands would only be tested for correct formatting. The actual string contents are not checked for correct spelling, etc.

1.2 Scope

The MiNET conformance test suite is based on MTS22C.

1.3 Background

This test suite uses valid and invalid frames to test the IUT behaviour. These terms are defined as follows:

Valid messages: A valid message is an message which is formatted as outlined in MTS22C and is supported by the IUT. Since not all MiNET messages are supported by every IUT, it is possible that a valid message for a particular IUT may be an invalid message for another IUT. As well, some IUTs do not support all defined parameters

within each MiNET message. These parameters will be, depending on the IUT, rejected or ignored.

Invalid messages: An invalid message is a message which:

is not part of MTS22C

is not formatted as outlined in MTS22C

is not supported by the IUT.

1.4 General Aspects

As per ISO/IEC 9646, “a complete and independent specification of the actions required to achieve a specific test purpose” is called an abstract test suite. These test cases, along with the test body, include a preamble and a postamble, which are defined below, to ensure starting and ending in a stable state and involve one or more consecutive or concurrent connections. However, as per ISO/IEC 9646 Part 2, it can be useful to use other stable states for starting and ending abstract test cases, in order to concatenate test cases in a manner which permits efficient execution.

1.5 Preamble

The preamble of a test case consists of the steps required to bring the IUT to the Normal (NM) state. The NM state is defined in MTS21.

1.6 Test Body

The test body is the sequence of steps within a test case, that is essential to achieve the test purpose. Verdicts are assigned to the possible outcomes of the test case.

1.7 Postamble

At the end of the execution of a test body, the IUT may not be in the Normal (NM) state. A postamble is required to return the IUT from its current state to the Normal (NM) state.

1.8 Timers

There are no timers associated with the MiNET protocol.

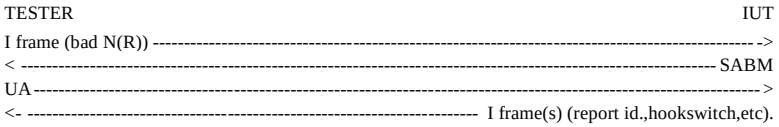
1.9 PICS/PIXIT Relationship to the Abstract Test Suite

There are instances when the execution of a test case depends on the answer to a PICS or PIXIT question. This makes the execution of the test case conditional.

2.0 Abstract test suite for MiNET

2.1 Preamble

The following preamble will be used for this ATS. This will ensure that the IUT is in the normal (NM) state.



```
RR ---------------------------------------------------------------------------------------------------->
```

2.2 Postamble

The following postamble will be used for this ATS. This will ensure that the IUT is in the SABM reset (RS) state.

```
TESTER                                                                                              IUT
I frame (bad N(R)) ------------------------------------------------------------------------------------>
< ------------------------------------------------------------------------------------------------ SABM
```

2.3 PICS/PIXIT Proforma

What is IUTs set Id. byte (hex)?

Does IUT have a display?

Does IUT have softkeys? If so, how many.What are the upstroke/downstroke key codes?

Does IUT have any hardkeys? If so, how many. What are the upstroke/downstroke key codes?

Does IUT have line indicators? If so how many.

Does IUT have a message lamp?

Does IUT have a microphone lamp?

Does IUT have handsfree capability?

What MiNET messages does IUT support?

Does IUT support MiLINK devices?

Set PBX Type

Date of testing

Limitations of IUT

Tester serial number

Test software version number

Tester operator's name

Does IUT support download procedure?

MTR options for the Request Maintenance Data command

2.4 Test cases for messages common to all terminals.

The following test cases are written using X- Notation. The Conformance tester sends the commands shown by the right arrows and the IUT sends the messages shown by the left arrows

2.4.1 Request Set ID (MT $C0) received by the IUT, Report Set ID (MT $C1) sent by the IUT

This test ensures that the IUT will report it's set id. to the tester. The tester checks the ID,CONFIG and PST bytes of the Report Set Id. message in accordance with the PIXIT settings.

```
TESTER                                                                                              IUT
Request Set Id. ---------------------------------------------------------------------------------------->
< ----------------------------------------------------------------------Report Set Id. (ID=,CONFIG=,PST=)
```

2.4.2 Network Level Message Reject (MT $C2) sent by the IUT.

This test verifies that the IUT will send MiNET message rejects to unknown or bad messages sent by the tester. The tester verifies the RMT byte of the MiNET Message Reject message. The RMT byte should contain the MT code of the unknown message. Since each IUT will only support a subset of the defined MiNET messages, the tester will use the PIXIT selections to determine what response is expected from the IUT.

```
TESTER                                                                    IUT
Illegal MT ($D7)) ------------------------------------------------------------------------------------------>
< ------------------------------------------------------------------------------ MiNET Message reject (RMT =D7)

FOR I = 1 TO FFx DO
   IF MT I IS UNSUNSUPPORTED
      All unsupported MT's ------------------------------------------------------------------------------------>
      <------------------------------------------------------------------------------ MiNET Message reject (RMT =)
   END
LOOP

Improperly formatted MiNET message ------------------------------------------------------------------------>
< ------------------------------------------------------------------------------ MiNET Message reject (RMT =)
MT with invalid parameters ---------------------------------------------------------------------------------->
< ------------------------------------------------------------------------------ MiNET Message reject (RMT =)
```

2.4.3 Read Set Memory (MT $C3) received by the IUT, Report Set Memory (MT $C4) sent by the IUT

This test verifies that the IUT will send a Report Set Memory message containing data found at the address specified in the Read Set Memory message sent by the tester. The number of data bytes returned is IUT dependant. This tester will look for the number of bytes specified in the PIXIT options.

```
TESTER                                                                    IUT
Read Set Memory (address = xxxx) ---------------------------------------------------------------------------->
< ------------------------------------------------------------------------------ Report Set Memory (data =)
```

2.4.4 Request Maintenance Data (MT $C5) received by IUT, Send Maintenance Data (MT $C6) sent by IUT

This test will ensure that the IUT can respond correctly to various MTR values. The PIXIT menu will allow the test operator to alter the default values. If the IUT does not support the MTR value sent, it should respond with the MRB byte set to $FF (not supported).

```
TESTER                                                                    IUT
Request Maintenance Data (MTR$01)--------------------------------------------------------------------------->
< ------------------------------------------------- Send Maintenance Data (MTR =$01, MRB = $01 passed)
                                                                  or
<------------------------------------------------------------ Send Maintenance Data (MTR = $01,
MRB = $FF)

Request Maintenance Data (MTR$02)--------------------------------------------------------------------------->
< -------------------------------------------------Send Maintenance Data (MTR =$02, MRB = $00 initiated)
                                                                  or
```

<---------- Send Maintenance Data (MTR = $02,
MRB = $FF)

Request Maintenance Data (MTR$03)---------->
< ----------Send Maintenance Data (MTR =$03, MRB = $00 initiated)
or
<---------- Send Maintenance Data (MTR = $03,
MRB = $FF)

Request Maintenance Data (MTR$04)---------->
< ----------Send Maintenance Data (MTR =$04, MRB = $00 initiated)
or
<---------- Send Maintenance Data (MTR = $04,
MRB = $FF)

Request Maintenance Data (MTR$05)---------->
< ----------Send Maintenance Data (MTR =$05, MRB = $00 initiated)
or
<---------- Send Maintenance Data (MTR = $05,
MRB = $FF)

Request Maintenance Data (MTR$06)---------->
< ---------- Send Maintenance Data (MTR =$06, MRB = $01 passed)
or
<---------- Send Maintenance Data (MTR = $06,
MRB = $FF)

Request Maintenance Data (MTR$07)---------->
< ---------- Send Maintenance Data (MTR =$07, MRB = $xx hardware stream)

<---------- Send Maintenance Data (MTR = $07,
MRB = $FF)

Request Maintenance Data (MTR$08)---------->
< ----------Send Maintenance Data (MTR =$08, MRB = $xx software Stream)
or
<---------- Send Maintenance Data (MTR = $08,
MRB = $FF)

Request Maintenance Data (MTR$09)---------->
< ---------- Send Maintenance Data (MTR =$09, MRB = $01 passed)
or
<---------- Send Maintenance Data (MTR = $09,
MRB = $FF)

Request Maintenance Data (MTR$0A1) ---------->
< ---------- Send Maintenance Data (MTR =$0A, MRB = $00 initiated)
or
<---------- Send Maintenance Data (MTR = $0A,
MRB = $FF)

Request Maintenance Data (MTR$0B) ---------->
< ----------Send Maintenance Data (MTR =$0B, MRB = $xx Tranducer register contents)
or
<---------- Send Maintenance Data (MTR = $0B,
MRB = $FF)

Request Maintenance Data (MTR$0C) ---------->
< ---------- Send Maintenance Data (MTR =$0C, MRB = $01 passed)
or
<---------- Send Maintenance Data (MTR = $0C,
MRB = $FF)

Request Maintenance Data (MTR$F1) ---------->

< ---------- Send Maintenance Data (MTR =$F1, MRB = $FF (not supported))

2.4.5 Request Set Reset (MT $C7) received by the IUT.

The tester sends a Request Set Reset command to the IUT. The IUT is expected to behave in the equivalent manner to power-up. That is, it should enter the SABM reset state. Voice sets should also (where applicable) turn DTMF tones on, turn all output transducers off, clear all delay updates, map buffer number 1 to the display, turn cursors off and put them at location (1,1), and use the B1 channel.

TESTER
IUT
Request Set Reset ---------->
< ----------RR
(The IUT may return RR prior to enterning SABM
Reset state.)

< ---------- SABM

2.4.6 Load Address (MT $C8) received by the IUT.

This message is not currently used by any sets.

2.4.7 Load Data (MT $C9) received by the IUT.

This message is not currently used by any sets.

2.4.8 Variable Length Load Data (MT $CA) received by the IUT.

This message is not currently used by any sets.

2.4.9 Checksum Request (MT $CB) received by the IUT.

This message is not currently used by any sets.

2.4.10 Checksum Reply (MT $CC) received by the IUT.

This message is not currently used by any sets.

2.4.11 Run (MT $CD) received by the IUT.

This message is not currently used by any sets.

2.4.12 Write Set Memory (MT $CE) received by the IUT.

This test will ensure that the IUT's memory can be modified using this command. The number of bytes of data that can be written via this command is IUT dependant. The tester will determine this value from the PIXIT options.

TESTER IUT

Write Set Memory (Start Addr=, Data=) ---------->
Read Set Memory (address = xxxx) ---------->
< ---------- Report Set Memory (data =)

2.4.13 Define Channel and MiLINK Parameters (MT $CF) received by the IUT.

This test will ensure that the IUT can correctly handle the parameters specified in the CHMB octet. The Define Channel and MiLINK Parameters command is used to control MiLINK locking, specify the B-channel that the IUT must use and to specify the data link layer address.

TESTER IUT
Define Channel and MiLINK Parameters (CHMB=)---------->
< ---------- Report Channel and MiLINK Parameters (CHMB =)
- check address
- check B-channel
- lock/unlock MiLINK

2.4.14 Configure Hardware (MT $D0) received by the IUT.

This test will verify that the IUT will react correctly to the Configure Hardware command. The Tester sends a Configure Hardware message with various parameter values. These parameters are IUT dependant and will be chosen according to PIXIT options.

TESTER IUT
Configure Hardware (DEV =1, ACT = 5)---------->
(disable volume keys)
Press
Volume Up Key
< ----------Send Key Code for Volume up key

Configure Hardware (DEV = 2, ACT = 3, LEN = 1, STREAM = 03)---------->
(setting codec Law and format)

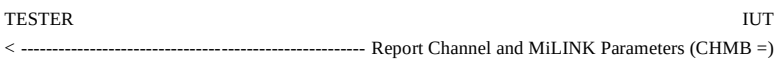
Configure Hardware (DEV = 3, ACT = 5) ---------->
(disable flashhook timing)
Take
IUT off hook
< ----------Send Key code for off hook
Perform
Flashhook
< ----------Send Key code for On hook
< ----------Send Key code for off hook
Put IUT
on hook.
< ---------- Send key code for on hook.

Configure hardware (DEV = 4,. ACT = 5)---------->
(Disable Ringing Frquency and Level)

Start Voice Set Ringer---------->
Check IUT
for silence.
Configure Hardware (DEV = 5, ACT = 5) ---------->
(Disabling Filter Coefficients)

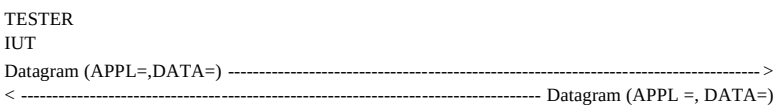
2.4.15 Report Channel and MiLINK Parameters (MT $D1) received by the IUT.

This test verifies that the IUT can correctly report B-channel selection and/or a MiLINK status changes. The test operator will be required to invoke some action on the IUT to cause it to send this message. One way to achieve this is to pull the DBUSY line low on the MiLINK device.



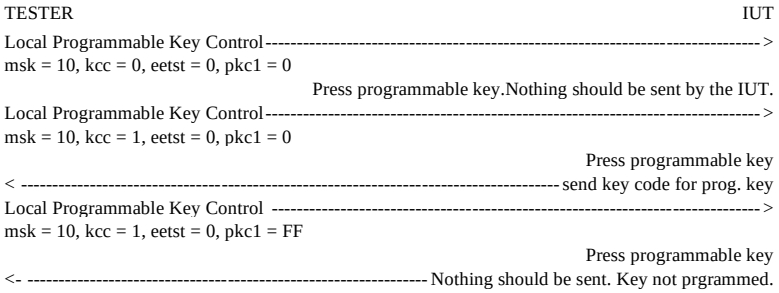
2.4.16 Datagram (MT $D2) received by the IUT.

This test verifies that the IUT can correctly respond to a Datagram message. This message can be sent in either direction.



2.4.17 Local Programmable Key Control (MT $D3) received by the IUT.

This test will verify that the IUT will respond correctly to the Local Programmable Key Control message contents sent by the tester. The tester first sends a Local Programmable Key Control command to the IUT. The contents of the message are such that the IUT should not report this key stroke to the tester when the key is depressed. The test operator is then instructed to press the key. The tester then sends another Local Programmable Key Control message to the IUT. This time the IUT is instructed to send the key code for the programmable key. The test operator is then prompted to press the key. The tester verifies that the correct key code is sent by the IUT. The tester then sends another Local Programmable Key Control message to the IUT. This time the IUT is instructed to enable key sequence reporting. The test operator is then prompted to press the programmable key. The tester verifies that the key sequence is sent by the IUT.



2.4.18 Miscellaneous Indicator Control (MT $D4) received by the IUT.

This test ensures the IUT will turn on/off/flash its miscellaneous indicator at the CRSR rate specified. Since each IUT may support different flash rates, the value for CRSR will be chosen from the PIXIT selections.

```
TESTER                                                                  IUT
Miscellaneous Indicator Control---------------------------------------------->
CRSR = $10 solid
IF RATE 1 SUPPORTED
                                                        ensure lamp is on solid.
ELSE
    <-----------------------------------------------------Network Reject (MT = $D4)
END


FOR I=2 TO 0Ex DO
  Miscellaneous Indicator Control---------------------------------------------->
  CRSR = Rate I
  IF RATE I SUPPORTED
                                                        ensure lamp is Flashing
  ELSE
    <-----------------------------------------------------Network Reject (MT = $D4)
  END
LOOP

Miscellaneous Indicator Control---------------------------------------------->
CRSR = $00 Flashing off
IF RATE 0 SUPPORTED
                                                                ensure lamp is off.
ELSE
    <-----------------------------------------------------Network Reject (MT = $D4)
END
```

2.4.19 Set Relay (MT $D5) received by the IUT.

This test ensures the IUT will activate/deactivate the specified relay(s).

```
TESTER                                                                  IUT
FOR I = 0 TO FFx DO
  IF RELAY I SUPPORTED
    Set Relay-------------------------------------------------------------------->
    RCTL= 1, RLY = I
                                                    ensure indicated relay is activated
    Set Relay-------------------------------------------------------------------->
    RCTL= 0, RLY = I
                                                  ensure indicated relay is deactivated
  END
LOOP
```

2.4.20 Pager Module Ringing (MT $D6) received by the IUT.

This test ensures the IUT will turn on/off it's ringer.

```
TESTER                                                                  IUT
Pager Module Ringing--------------------------------------------------------->
PMCTL= 01
                                                                ensure ringer is on
Pager Module Ringing--------------------------------------------------------->
PMCTL= 00
```

ensure ringer is off

2.5 Voice sets and console message (system to terminal) test cases

2.5.1 Send String (MT $01) received by the IUT.

This test verifies that an IUT with a display can store the string found in the Send String command in the requested position. The tester first sends a Display Block Cursor Command to allow the test operator to see the present cursor position. The tester then sends a Request Cursor Position command to the IUT. The IUT is expected to return its present cursor position. The tester then sends the Send String command with a few different BCCB bytes and checks that the IUT responds correctly to the return cursor bit. The test operator is responsible for checking that the correct string appears in the correct position on the IUT.

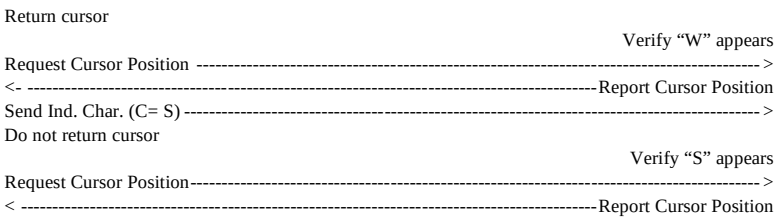
BCCB: bit 8 = return cursor. (0 = do not return cursor, 1 = return cursor to where it was prior to operation)
bit 7 = (Y,X) or Cursor number (0 = use (Y,X), 1 = use cursor number)
bits 6,5,4 = Cursor number (values 0 to 7 = cursor numbers 1 to 8)
bits 3,2,1 = Buffer Number (0 = all buffers, Values 1 to 7 = buffers 1 to 7)

TESTER IUT
Initialize Dsiaply ---------->
Display Underscore Cursor ---------->
Request Cursor Position---------->
< ----------Report Cursor Position
Send String (Y=1,X=3,NC=A, string = Hello there) ---------->
return cursor
Verify “Hello there” appears
Request Cursor Position---------->
< ----------Report Cursor Position
Send String (NC=8, string = Nice Day)---------->
do not return cursor
Verify “Nice Day” appears
Request Cursor Position ---------->
< ----------Report Cursor Position
Send String (Y=2,X=1,NC=D, string =**C0**Hello there**C1**)---------->
Verify “Hello there” appears flashing

2.5.2 Send Individual Character (MT $02) received by the IUT.

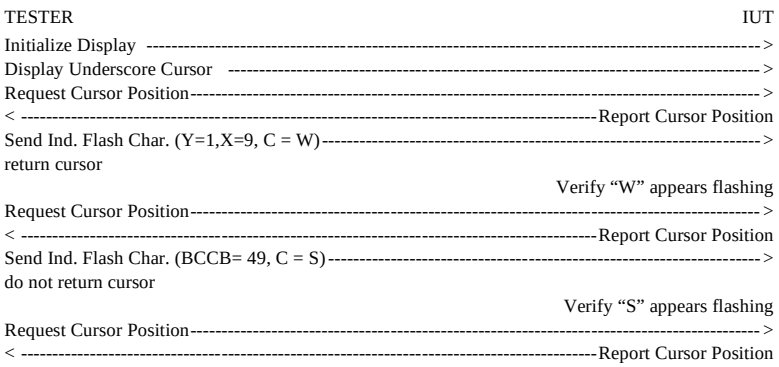
This test verifies that an IUT with a display can store the character found in the Send Individual Character command in the requested position. The tester first sends commands to display a block cursor and report cursor position to the IUT. The tester then sends Send Individual Character commands with a few different BCCB values. The tester verifies that the return cursor bit is acted on correctly by the IUT. The test operator is responsible for verifying that the IUT displays the correct characters in the correct position.

TESTER IUT
Initialiaze Display ---------->
Display UnderscroreCursor ---------->
Request Cursor Position---------->
< ----------Report Cursor Position
Send Ind. Char. (Y=1, X=3, C= W)---------->



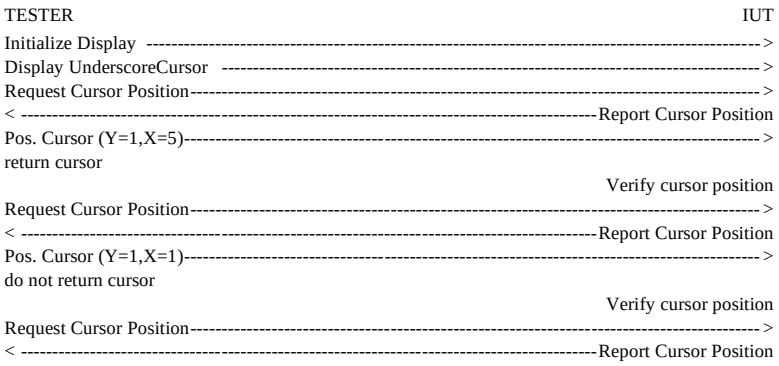
2.5.3 Send Individual Flashing Character (MT $03) received by the IUT.

This test verifies that an IUT with a display can store the character found in the Send Individual Flashing Character command in the requested position. This test is similar to the previous test except that this command flashes the displayed character.



2.5.4 Position Cursor (MT $04) received by the IUT.

This test verifies that an IUT with a display can move the cursor to the position specified in the Position Cursor command. This tester first sends a command to display the block cursor in order for the test operator to see its current position. The tester then requests the current position of the cursor from the IUT. The tester then sends the Position Cursor command with a few different BCCB values and verifies that the IUT responds correctly. The test operator is responsible for verifying that the IUT moves its cursor to the requested position on the display.



2.5.5 Shift Display Field Left (MT $05) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, shift its displays position one character position to the left and add a new character to the right. After execution of this command the cursor should be returned to the position it occupied before this command was received. Note that the field to be shifted must be a subset of one line on the display. The tester first determines the cursor position and then sends the string Brow to the IUT. The tester then sends the character n flashing. The tester then sends the Shift Display Field Left command. The IUTs display should now read “rownd” with the n flashing. The tester then sends another Shift Display Field Left command except this time it uses a different BCCB value and the character z. The display should now be “owndz” with the n flashing.The tester verifies that the IUT responded correctly to the Request Cursor Position commands. The test operator is required to verify that the IUTs display is shifted correctly.

```
TESTER                                                                  IUT
Initialize Display  ---------------------------------------------------------------------------->
Display Underscore Cursor (BCCB= 49)----------------------------------------------------------->
Request Cursor Position------------------------------------------------------------------------->
< ----------------------------------------------------------------------------Report Cursor Position
Send String (Y=1,X=3,NC=4, string = Brow)------------------------------------------------------>
Send Ind. Flash Char. (BCCB= 49, C = n)-------------------------------------------------------->
Shift Display Field Left (Y=1,X=3,FL=5,C=d) ---------------------------------------------------->
use Y,X coordinates
                                                        Verify display and cursor position
Request Cursor Position------------------------------------------------------------------------->
<- ---------------------------------------------------------------------------Report Cursor Position
Position Cursor ( Y=1, X=3) -------------------------------------------------------------------->
Shift Display Field Left (FL=5,C=z)  ----------------------------------------------------------->
use current cursor position
                                                        Verify display and cursor position
Request Cursor Position------------------------------------------------------------------------->
< ----------------------------------------------------------------------------Report Cursor Position
Shift Display Field Left (FL=5,C=z)------------------------------------------------------------>
                                                  Display should not change due to BCCB
```

2.5.6 Shift Display Field Right (MT $06) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, shift its displays position one character position to the right and add a new character to the left. After execution of this command the cursor should be returned to the position it occupied before this command was received. Note that the field to be shifted must be a subset of one line on the display. The tester first determines the cursor position and then sends a Send Individual Flashing Character followed by a Send String. The tester then sends the Shift Display Field Right command with a few different BCCB bytes. The tester verifies that the IUT responded correctly to the commands. The test operator is required to verify that the IUTs display is shifted correctly.

```
TESTER                                                                  IUT
Initialize Display  ---------------------------------------------------------------------------->
Display Underscore Cursor  --------------------------------------------------------------------->
Request Cursor Position------------------------------------------------------------------------->
< ----------------------------------------------------------------------------Report Cursor Position
Send Ind. Flash Char. (BCCB= 49, C= B) --------------------------------------------------------->
Send String (Y=1,X=2,NC=4, string = rown)------------------------------------------------------>
```

Shift Display Field Right (Y=1,X=2,FL=5,C=k)---------->
use Y,X coordinates
Verify display and cursor position
Request Cursor Position---------->
<----------Report Cursor Position
Position Cursor (Y=1, X=2) ---------->
Shift Display Field Right (FL=5,C=7)---------->
use current cursor position
Verify display and cursor position
Request Cursor Position---------->
<----------Report Cursor Position
Shift Display Field Right (FL=5,C=7)---------->
Display should not change due to BCCB

2.5.7 Roll Display Field Left (MT $07) received by the IUT.

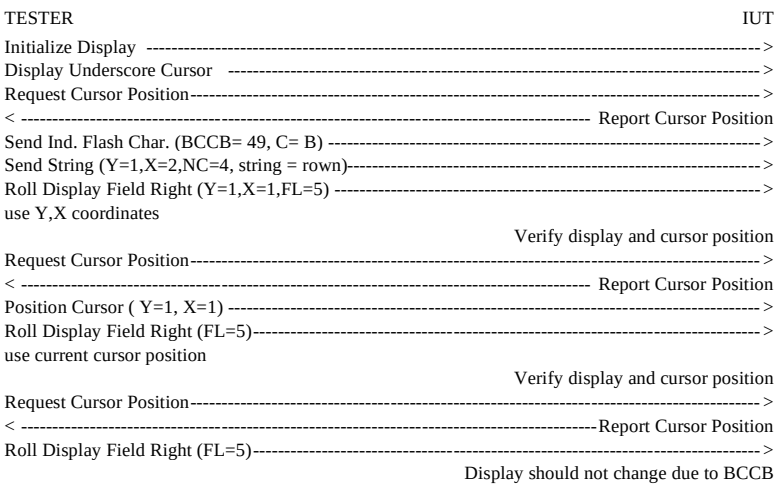
This test verifies that an IUT can, upon receipt of this command, roll its displays position one character position to the left. The character at the left end of the string prior to the operation, is rolled into the right-most position after the operation. After execution of this command the cursor should be returned to the position it occupied before this command was received. Note that the field to be rolled must be a subset of one line on the display. The tester first determines the cursor position and then sends the Roll Display Field Left command with a few different BCCB bytes. The tester verifies that the IUT responded correctly to the commands. The test operator is required to verify that the IUTs display is shifted correctly.

TESTER IUT
Initialize Display ---------->
Display Underscore Cursor ---------->
Request Cursor Position---------->
<----------Report Cursor Position
Send String (Y=2,X=3,NC=4, string = Brow)---------->
Send Ind. Flash Char. (BCCB= 49, C= n)---------->
Roll Display Field Left (Y=1,X=7,FL=5) ---------->
use Y,X coordinates
Verify display and cursor position
Request Cursor Position---------->
<----------Report Cursor Position
Position Cursor (Y=1, X=3) ---------->
Roll Display Field Left (FL=5) ---------->
use current cursor position
Verify display and cursor position
Request Cursor Position---------->
<----------Report Cursor Position
Roll Display Field Left (FL=5) ---------->
Display should not change due to BCCB

2.5.8 Roll Display Field Right (MT $08) received by the IUT.

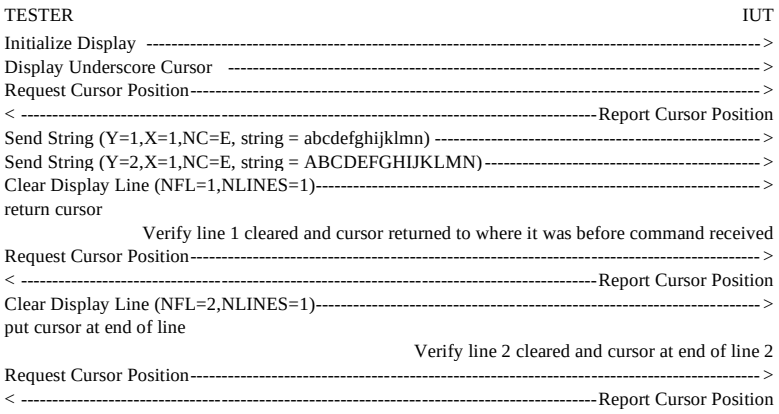
This test verifies that an IUT can, upon receipt of this command, roll its displays position one character position to the right. The character at the right end of the string prior to the operation, is rolled into the left-most position after the operation. After execution of this command the cursor should be returned to the position it occupied before this command was received. Note that the field to be rolled must be a subset of one line on the display. The tester first determines the cursor position and then sends the Roll Display Field Left command with a few different BCCB

bytes. The tester verifies that the IUT responded correctly to the commands. The test operator is required to verify that the IUTs display is shifted correctly.



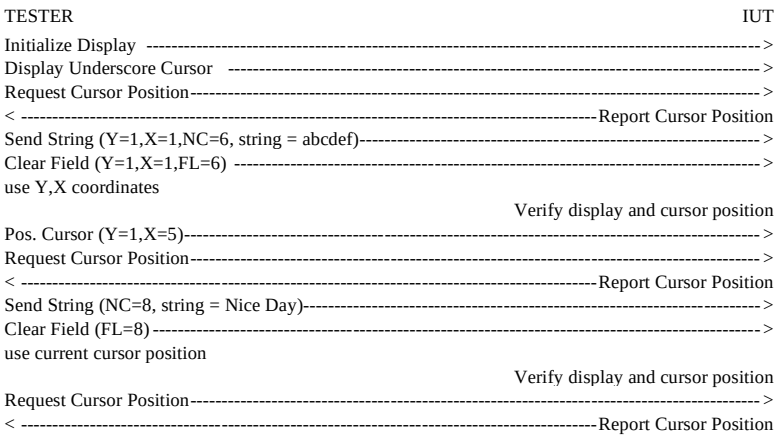
2.5.9 Clear Display Line (MT $09) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, clear the specified lines on its display. If the cursor return bit in the BCCB is set to 1, the cursor shall remain in the position in occupied prior to the operation. If the return cursor bit is 0, the cursor should move to the left-most position of the last line cleared by the operation.



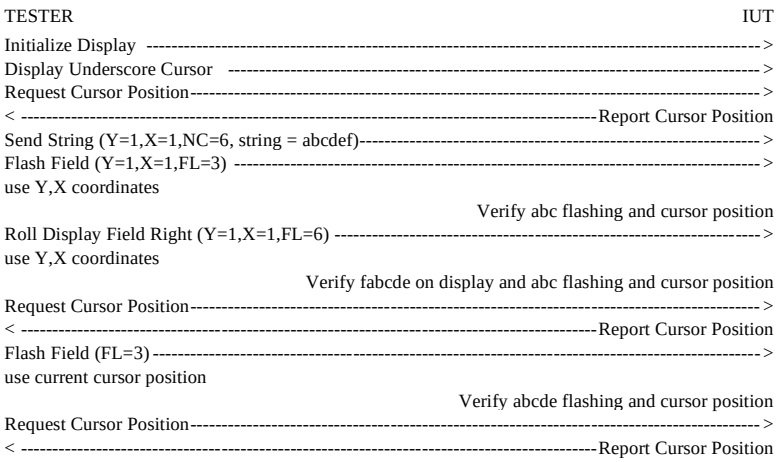
2.5.10 Clear Field (MT $0A) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, clear the specified field on its display. The tester first sends a command to display the cursor. It then requests the current cursor position. The tester then sends a Clear Field command using various BCCB values. The test operator is responsible for checking that the IUTs display responds properly.



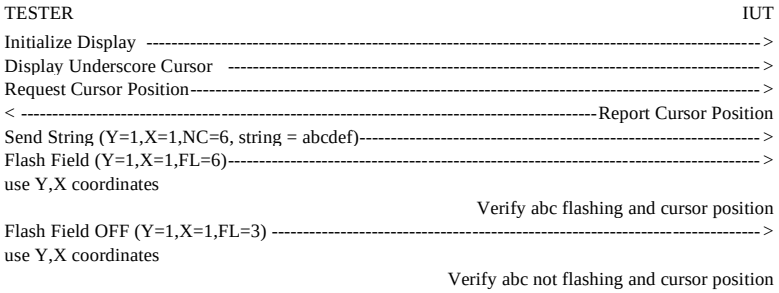
2.5.11 Flash Field ON (MT $0B) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, flash the specified field on its display.



2.5.12 Flash Field OFF (MT $0C) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, disable flashing of the specified field on its display.



Request Cursor Position---------->
<- ----------Report Cursor Position
Flash Field OFF (FL=3)---------->
use current cursor position
Verify abcdef not flashing and cursor position
Request Cursor Position ---------->
<- ----------Report Cursor Position

2.5.13 Display Block Cursor (MT $0E) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, cause a block cursor to be displayed at the current cursor position.

TESTER IUT
Initialize Display ---------->
Pos. Cursor (Y=1,X=5)---------->
Display Block Cursor ---------->
Verify display and cursor position
Request Cursor Position---------->
< ----------Report Cursor Position

2.5.14 Display Underscore Cursor (MT $0F) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, cause an underscore cursor to be displayed at the current cursor position.

TESTER IUT
Initialize Display ---------->
Pos. Cursor (Y=1,X=9)---------->
Display Underscore Cursor ---------->
Verify display and cursor position
Request Cursor Position---------->
< ----------Report Cursor Position

2.5.15 Blank Cursor (MT $10) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, cause a blank cursor to be displayed at the current cursor position.

TESTER IUT
Initialize Display ---------->
Pos. Cursor (Y=1,X=9)---------->
Display Underscore Cursor ---------->
Display Blank Cursor ---------->
Verify display and cursor position
Request Cursor Position---------->
< ----------Report Cursor Position

2.5.16 Copy Buffer Line (MT $11) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, copy a line from one display buffer to another.The tester stores a string in buffer 2 line1. Then the tester sends the Copy Buffer Line command requesting that the IUT copy line 1 from buffer 2 and put it in line 2 of buffer 1. Since buffer 1 is currently being displayed the string should appear on line 2 of the display.

TESTER IUT
Initialize Display ---------->

Display Underscore Cursor ---------->
Send String (NC=4, string = copy) ---------->
Copy Buffer Line (SBDB=11,YSRC=01,YDEST=02) ---------->
Verify copy appears on line 2 of display and cursor position

2.5.17 Clear Screen (MT $12) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, clear its display and places its cursor in the top left corner of the display. If the IUT supports prompts, then these shall be deactivated.

TESTER IUT
Initialize Display ---------->
Display Underscore Cursor ---------->
Send String (Y=1,X=5,NC=4, string = copy)---------->
Request Cursor Position---------->
< ----------Report Cursor Position
Clear Screen ---------->
Verify display and cursor position
Display Underscore Cursor ---------->
Request Cursor Position---------->
< ----------Report Cursor Position

2.5.18 Delay Update of a Field (MT $13) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, wait a specified length of time before updating its display. The tester sends a Delay Update of a Field command specifying a 1 second delay of the string “zyxwvut. The tester then sends the string “abc” requesting a 2 second delay to update.

TESTER IUT
Initialize Display ---------->
Display Underscore Cursor ---------->
Request Cursor Position---------->
< ----------Report Cursor Position
Send String (NC=4, string = copy) ---------->
Delay Update of a Field (Y=1,X=1,FL=7,T=14,N=7,C=zyxwvut)---------->
Using Y,X Coordinates
Verify display and cursor position
Request Cursor Position---------->
< ----------Report Cursor Position
Position Cursor (Y=1, X=1) ---------->
Delay Update of a Field (FL=4,T=28,N=3,C=abc) ---------->
Using current cursor position
Verify display and cursor position
Request Cursor Position---------->
< ----------Report Cursor Position

2.5.19 Display Prompt (MT $14) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, display previously stored text in a soft key area.

TESTER IUT
Initialize Display ---------->
Store Prompt (PV=6,NC=7,C=prompt0) ---------->
Store Prompt (PV=7,NC=7,C=prompt1) ---------->
Display Prompt (NPBN=11,PV1=06,PV2=07)---------->
Verify display

Display Prompt (NPBN=09,PV1=FE) ---------->
254PV (blank prompt position)
Verify display
Display Prompt (NPBN=09,1PV=FD) ---------->
253PV (deactivate prompt)
Verify display
Display Prompt (NPBN=09,1PV=FF) ---------->
255PV (no change to prompt position)
Verify display

2.5.20 Store Prompt (MT $15) received by the IUT.

Already tested in § 2.5.19.

2.5.21 Send Prompt (MT $16) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, immediately display a prompt which is not pre-stored in the IUT's memory.

TESTER IUT
Initialize Display ---------->
Send Prompt (PPBN=09,NC=4,C=new1) ---------->
Verify display
Send Prompt (PPBN=09,NC=1A,C=abcdefghijklmnopqrstuvwxyz) ---------->
NC is too long so IUT should truncate it.
Verify display

2.5.22 Deactivate Prompt (MT $17) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, deactivate a specified prompt, including any associated delimiters and boxes. A send string command is then sent to verify that the area is usable. Then a Display Prompt command is sent to verify the prompt area including delimiters and boxes can be set.

TESTER IUT
Initialize Display ---------->
Store Prompt (PV=6, NC=7,C=prompt0)---------->
Store Prompt (PV=7, NC=7, C=prompt1)---------->
Display Prompt (NPBN=11,PV1=06,PV2=07)---------->
Deactivate Prompt (PPBN=09) ---------->
Verify display
Send String (Y=3,X=1,NC=4,C=help)---------->
Verify display
Display Prompt (NPBN=09,PV1=06)---------->
Verify display

2.5.23 Display Buffer (MT $18) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, display the specified internal buffer.

TESTER IUT
Initialize Display ---------->
Display Underscore Cursor ---------->
Send String (NC=5, string = buff2)---------->
Display Buffer ---------->
Verify display and cursor position

Display Buffer >

2.5.24 Set LCD Indicator (MT $19) received.

This test turns on the LCD indicators or LED indicators depending on the IUT type. The PIXIT options will determine what CRSR and IN values will be sent to the IUT. The tester also checks that various invalid values of CRSR and IN are rejected.

TESTER IUT

Set LCD indicator (CRSR = 1x, IN = $xx)----------->
($xx; $01=LED, $FF=LCD)

Ensure LED/LCD is on. solid

FOR i = 1 to 05x DO

Set LCD indicator (CRSR = $yy, IN = $xx) ----------->
($yy; **I**=LED, **II**=LCD - $xx; $01=LED, $FF=LCD)

Ensure LED/LCD is Flashing

or

<---------- Network Level Reject(RMT=19)

LOOP

FOR i = 6 to Cx DO

IF RATE SUPPORTED

Load Flash Rate (CRSN=**i,** OOT=AAx)----------->
($yy; **I**x=LED, **II**x=LCD)

Set LCD indicator (CRSR = $yy, IN = $xx) ----------->
($yy; **I**x=LED, **II**x=LCD - $xx; 01x=LED, FFx=LCD)

Ensure LED/LCD is Flashing

ELSE

Set LCD indicator (CRSR = $yy, IN = $xx) ----------->
($yy; **I**x=LED, **II**x=LCD - $xx; 01x=LED, FFx=LCD)

<---------- Network Level Reject(RMT=19)

END

LOOP

Set LCD indicator (CRSR = $yy, IN = $xx) ----------->
($yy; 0Ex=LED, EEx=LCD - $xx; 01x=LED, FFx=LCD)

Ensure LED/LCD is Flashing.

or

< ---------- Network Level Reject(RMT=19)

IF RATE_6 SUPPORTED

Load Flash Rate (CRSN=$yy, OOT=AAx) ----------->
($yy; 06x=LED, 66x=LCD)

Set LCD indicator (CRSR = $yy, IN = $xx) ----------->
($yy; 06x=LED, 66x=LCD - $xx; 01x=LED, FFx=LCD)

Ensure LED/LCD is Flashing

Set LCD indicator (CRSR = $yy, N = $xx) ----------->
($yy; 06x=LED, 66x=LCD - $xx; 02x=LED, FEx=LCD)

Ensure LED/LCD is still Flashing

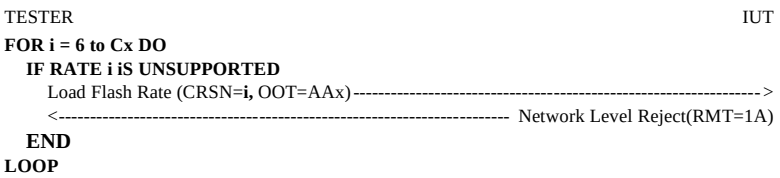
END

Set LCD indicator (CRSR = 00x, IN = $xx) ----------->
($xx; 01x=LED, FFx=LCD)

Ensure LED/LCD is off.

2.5.25 Load flash rate (MT = $1A) received by the IUT.

This test verifies that an IUT can load various flash rates into it's memory. The flash rates supported are IUT dependant. The tester sends the various flash rates selected in the PIXIT menu. Some unsupported flash rates will also be sent to ensure the IUT rejects them.

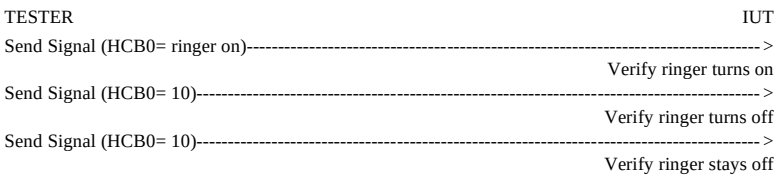


2.5.26 Send Signal (MT $1B) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, control the various hardware functions specified. Since the various parameters are set dependant, the tester will choose HCB0 and HCB1 values from PIXIT selections.

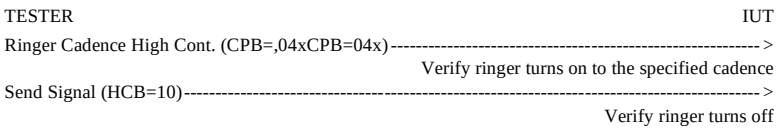
HCB0: bit 8 = on/off
bit 7 = 0
bit 6 = HS (handset microphone) Controls handset microphone, earpiece and sidetone together.
bit 5 = HFS (handsfree speaker)
bit 4 = HFMF (handsfree microphone function)
bit 3 = DG (DTMF generator)
bit 2 = NC (noise cancellation during handsfree operation)
bit 1 = RNG (ringer)

HCB1: bit 8 = on/off
bit 7 = 1
bit 6 = 0
bit 5 = HFM (handsfree microphone)
bit 4 = LIND (line DTMF)
bit 3 = HSNV (handset nominal volume)
bit 2 = SIDE (sidetone)
bit 1 = EAR (earpiece)



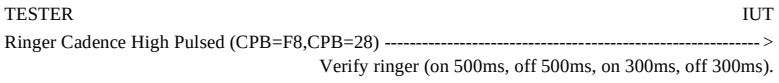
2.5.27 Ringer Cadence High Continuous (MT $1C) received by the IUT.

This test verifies that an IUT (currently consoles only) can, upon receipt of this command, start its ringer using the specified cadence. This pattern should continue until turned of by a Send Signal command with the RNG bit turned off.



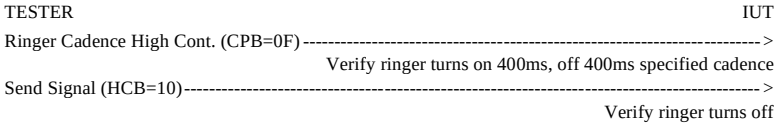
2.5.28 Ringer Cadence High Pulsed (MT $1D) received by the IUT.

This test verifies that an IUT (currently consoles only) can, upon receipt of this command, start its ringer using the specified cadence. Only one cadence cycle should be generated.



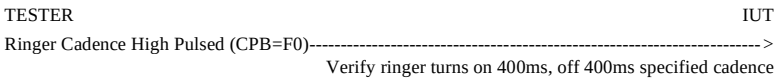
2.5.29 Ringer Cadence Low Continuous (MT $1E) received by the IUT.

This test verifies that an IUT (currently consoles only) can, upon receipt of this command, start its ringer using the specified cadence. This pattern should continue until turned off by a Send Signal command with the RNG bit turned off.



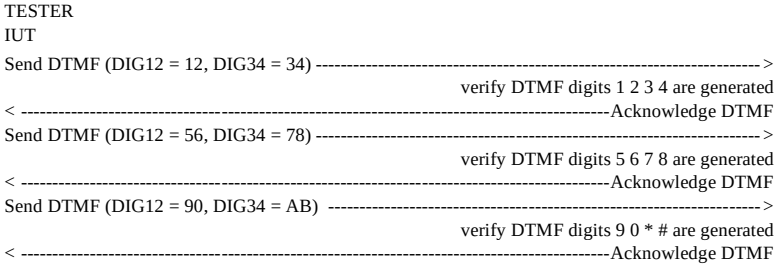
2.5.30 Ringer Cadence Low Pulsed (MT $1F) received by the IUT.

This test verifies that an IUT (currently consoles only) can, upon receipt of this command, start its ringer using the specified cadence. Only one cadence cycle should be generated.



2.5.31 Send DTMF (MT = $20) received by the IUT.

This should cause the IUT to generate four DTMF tones with the on/off time of 100ms/100ms. In the send DTMF parameters, hex digits 0 through B will cause sending of DTMF tones 0 - 9,*,#. Hex digits C - F will cause 100 ms of silence.Check that this happens on the set



Send DTMF (DIG12 = CD, DIG34 = EF)---------->
verify silence is generated
< ----------Acknowledge DTMF
Send DTMF (DIG12 = 12, DIG34 = 34) ---------->
press a digit on the IUT keypad
verify digit dialling is aborted and no Ack DTMF sent

2.5.32 Request Cursor Position (MT $22) received by the IUT.

Already tested in § 2.5.1 to § 2.5.15.

2.5.33 Start Precise Ringer Cadencing (MT = $23) received by the IUT.

This test verifies that the IUT's ringer outputs the correct cadence.

TESTER IUT
Start precise ringer cadencing (CPB =AC, RP =9A, CPP=84) ---------->
Pulsed
0n 400ms, off 200ms, on 400ms, off 200ms, on 800ms, off 4 s
check ringer
Start precise ringer cadencing (CPB =AC, RP =1A, CPP=84)---------->
Continuous
Ensure ringer is outputting correct pattern
Send SETUP (turn ringer off) ---------->
Verify ringer turns off

2.5.34 Set Codec Gain (MT = $24) received by the IUT

This test ensures that the IUT will correctly adjust its codec's gain as per the parameters in the Set Codec Gain message. The GAIN parameters are terminal dependant and will be a PIXIT option.

TESTER IUT
Set codec gain (GAIN =77) ---------->
check on handset
Set codec gain (GAIN =00)---------->
check on handset

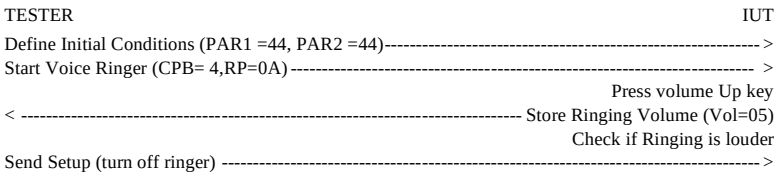
2.5.35 Start Voice Set Ringer (MT = $26) received by the IUT.

This test ensures that the IUT's ringer will output the cadence specified in the Start Voice Set Ringer command.

TESTER IUT
Start Voice Set Ringer (CPB =AC, RP =9A) ---------->
Pulsed
0n 200ms, off 200ms, on 200ms, off 200ms, on 400ms, off 400ms
check ringer
Start Voice Set Ringer (CPB =AC, RP =1A) ---------->
Continuous
0n 200ms, off 200ms, on 200ms, off 200ms, on 400ms, off 400ms
check ringer
Send SETUP (turn ringer off) ---------->
Verify ringer turns off

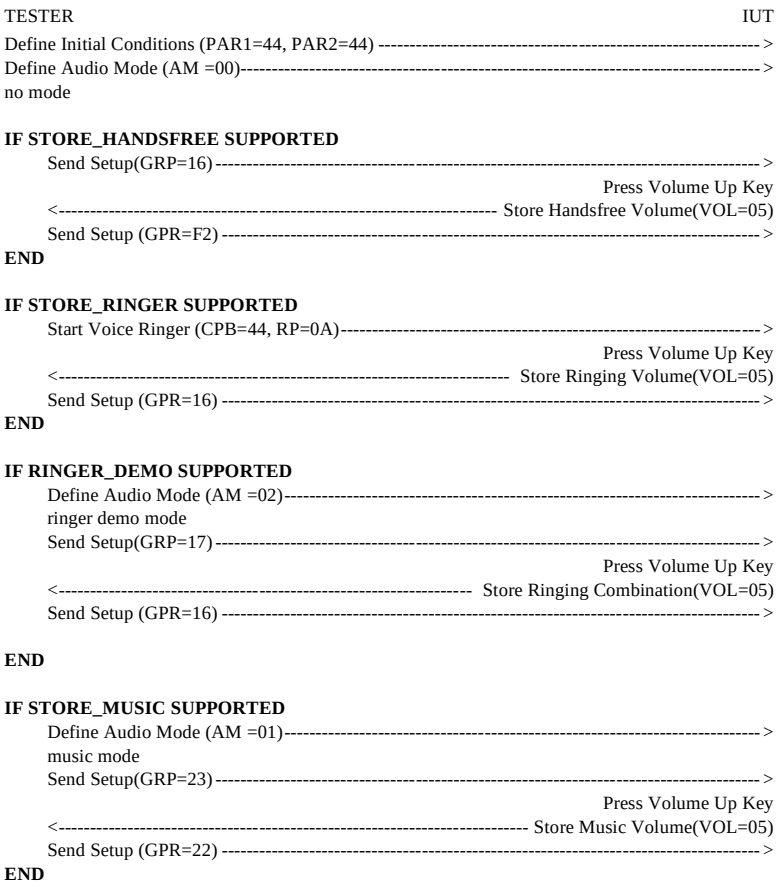
2.5.36 Define initial Conditions (MT= $27) received by the IUT.

This test verifies that the IUT correctly stores the initial parameters as defined in the PAR1 and PAR2 bytes. Each IUT supports different parameter combinations and therefore the tester will choose these parameters from the PIXIT setup.



2.5.37 Define Audio mode (MT = $28) received by IUT.

This test uses the PIXIT parameters to send the various supported AM values to the IUT. The test operator must ensure the correct operation.

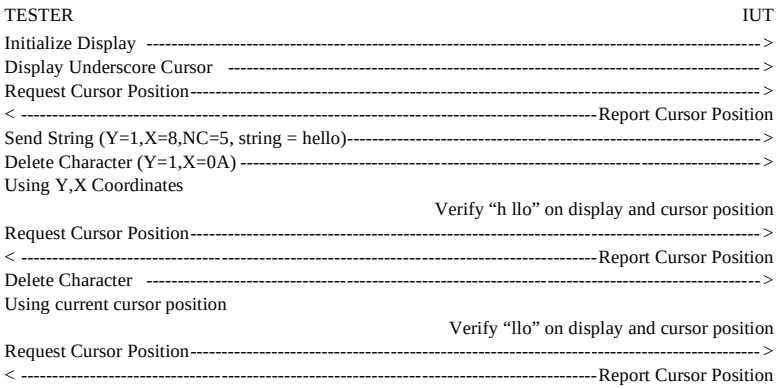


2.5.38 Send Data Module Command (MT $29) received by the IUT.

This command is for data modules only. This suite does not currently support data modules.

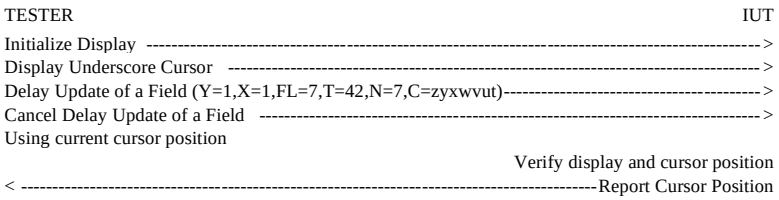
2.5.39 Delete Character (MT $2A) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, move the specified cursor back one position, and blank the character previously at that position.



2.5.40 Cancel Delay Update Of Field (MT $2B) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, cancel the pending delayed update of a field. The Delay Update of a Field command is issued first. Then the Cancel command is issued. The IUT should NOT display the string sent in the Delay Update of a Field command.



2.5.41 Initialize Display (MT $2C) received by the IUT.

This test verifies that an IUT (which has internally-defined prompts) can, upon receipt of this command, initialize its display. The IUT should clear the designated text lines in the specified buffer, return the specified cursor to the “home” position and cancel any pending delayed updates for the specified buffer. Note that any designated prompt lines are not affected by this command. The tester first establishes a delayed update of a field. The tester then sends the Initialize Display command to the IUT. The IUT should initialize its text lines and cancel the delayed update.

Delay Update of a Field (Y=1,X=1,FL=7,T=42,N=7,C=zyxwvut)---------->
Initialize Display ---------->
ensure text lines erased, prompts not erased, cursor in home position and delay update cancelled
< ----------Report Cursor Position

2.5.42 Select Stored Prompt (MT $2D) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, display previously stored text in a soft key area.

TESTER IUT
Initialize Display ---------->
Store Prompt (PV=6,NC=7,C=prompt0) ---------->
Store Prompt (PV=7,NC=7,C=prompt1) ---------->
Select Stored Prompt (NPBN=11,FPP=2,1PV=06,2PV=07)---------->
Verify display
Select Stored Prompt (NPBN=09,FPP=3,1PV=FE) ---------->
254PV (blank prompt position)
Verify display
Select Stored Prompt (NPBN=09,FPP=2,1PV=FD ---------->
253PV (deactivate prompt)
Verify display
Select Stored Prompt (NPBN=09,FPP=2,1PV=FF)---------->
255PV (no change to prompt position)
Verify display

2.5.43 Define 5x7 Special Character (MT $2E) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, define a special character using a 5x7 dot matrix.

TESTER IUT
Initialize Display ---------->
Define 5x7 Special Character (CC=A2,PB0=7F,PB1=09,PB2=09,PB3=09,PB4=01) ---------->
Send String (NC=1, string = A2) ---------->
Verify special character

2.5.44 Define Character Pattern (MT $30) received by the IUT.

This test verifies that an IUT can, upon receipt of this command, define a special character.

TESTER IUT
Initialize Display ---------->
Define Character Pattern (A2=80,DX=08,DY=08,STREAM=3e,20,20,3c,20,20,20,00) ---------->
Send String (NC=1, string = A2) ---------->
Verify special character

2.5.45 Setup (MT = $33) received by IUT.

This test verifies all the possible combinations of parameters contained in the SETUP message. Some IUT's will only support a subset of the parameters. The tester will choose the parameters from the PIXIT selections.

GRP0: bit 8 = SIDE (sidetone)
bit 7 = HSNV (handset nominal volume (restore))

bit 6 = HSS (handset speaker)
bit 5 = HSM (handsfree microphone)
bit 4 = 0
bit 3 = 0
bit 2 = 0
bit 1 = ON/OFF

GRP1: bit 8 = LPF (low-pass filter)
bit 7 = HFA (handsfree algorithm)
bit 6 = RSS (set speaker)
bit 5 = RSM (set microphone)
bit 4 = 0
bit 3 = 0
bit 2 = 1
bit 1 = ON/OFF

GRP2: bit 8 = x
bit 7 = x
bit 6 = DTRX (DTMF receive (to local transducers))
bit 5 = DTTX (DTMF transmit (to system via PCM))
bit 4 = 0
bit 3 = 1
bit 2 = 0
bit 1 = ON/OFF

GRP3: bit 8 = x
bit 7 = x
bit 6 = x
bit 5 = RING (ringer)
bit 4 = 0
bit 3 = 1
bit 2 = 1
bit 1 = ON/OFF

GRP4: bit 8 = LPF (low-pass filter)
bit 7 = HFA (handsfree algorithm)
bit 6 = RSS (remote set speaker)
bit 5 = RSM (remote set microphone)
bit 4 = 1
bit 3 = 0
bit 2 = 0
bit 1 = ON/OFF

GRPT = $0D

TESTER IUT
Setup (GRP0 =, GRP1 =, GRP2 =,GRP3 =,GRP4 =, GRPT) ---------->
Check what happens if no terminator byte

2.5.46 Start New Call Tone (MT = $34) received by the IUT.

This test ensures that the IUT's ringer will output the cadence specified in the Start New Call Tone command.

```
TESTER                                                                  IUT
Start New Call Tone (CPB =AC, RP =9A)---------------------------------------------->
Pulsed
0n 200ms, off 200ms, on 200ms, off 200ms, on 400ms, off 400ms
                                                                  check ringer
Start New Call Tone (CPB =AC, RP =1A) ---------------------------------------------->
Continuous
0n 200ms, off 200ms, on 200ms, off 200ms, on 400ms, off 400ms
                                                                  check ringer
Send SETUP (turn ringer off) ------------------------------------------------------->
                                                        Verify ringer turns off
```

2.6 Voice sets and console messages (terminal to system) test cases

2.6.1 Send QWERTY Key (MT $01) received by the IUT.

This test verifies that the IUT (which has a QWERTY keyboard) can correctly report an upstroke or a downstroke of a keyboard key.

Not automated in this release.

2.6.2 Send Scan Key (MT $02) received by the IUT.

This test ensures that each key reports the proper message to the switch. For dial pad keys 0-9,*,# only the upstroke is reported. The hard keys report both up and down stroke. Each IUT supports different keys. The tester will prompt the test operator to send the keys as per the PIXIT selections.

```
TESTER                                                                  IUT
IF DIAL_KEYPAD SUPPORTED
    <---------------------------------------------------------------------- Send scan key(dial 1)
    <---------------------------------------------------------------------- Send scan key(dial 2)
    <---------------------------------------------------------------------- Send scan key(dial 3)
    <---------------------------------------------------------------------- Send scan key(dial 4)
    <---------------------------------------------------------------------- Send scan key(dial 5)
    <---------------------------------------------------------------------- Send scan key(dial 6)
    <---------------------------------------------------------------------- Send scan key(dial 7)
    <---------------------------------------------------------------------- Send scan key(dial 8)
    <---------------------------------------------------------------------- Send scan key(dial 9)
    <---------------------------------------------------------------------- Send scan key(dial 0)
    <---------------------------------------------------------------------- Send scan key(dial *)
    <---------------------------------------------------------------------- Send scan key(dial #)
END

IF HANDSET SUPPORTED
    <---------------------------------------------------------------------- Send scan key (Off hook)
    <---------------------------------------------------------------------- Send scan key (On hook)
END

FOR i = 1 to 1Ex DO
  IF LINE_KEY i SUPPORTED
    <---------------------------------------------------------------------- Send scan key(Press Line KEY i)
  END
LOOP

FOR i = 1 to 6 DO
  IF SOFT_KEY i SUPPORTED
    <---------------------------------------------------------------------- Send scan key(Press SOFT_KEY i)
```

END
LOOP

2.6.3 Acknowledge DTMF (MT $03) received by the IUT.

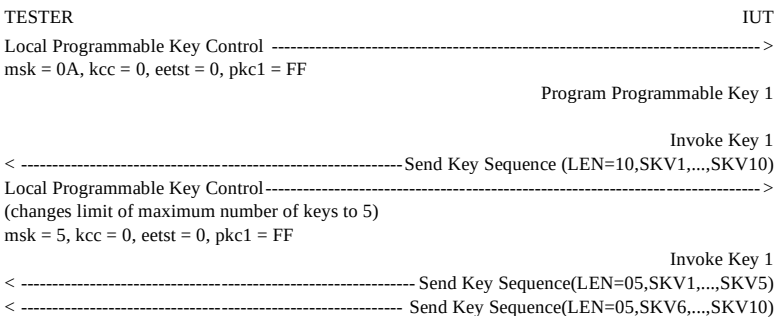
Already tested in § 2.5.31.

2.6.4 Report Cursor Position (MT $04) received by the IUT.

Already tested in § 2.5.1 to §2.5.15.

2.6.5 Send Key Sequence (MT $05) received by the IUT.

This test ensures that the set will correctly transmit this message to the tester. The user is requested to send a send key sequence command by pressing a speed call key. The IUT should send a maximum of 10 digits (default). The tester then limits the length of the send Key Sequence command by sending a Local Programmable Key Control message. The test operator then presses the speed call key and this time only 5 digits should be sent.



2.6.6 Store Ring Combination (MT $07) received by the IUT.

The tester sends the IUT a define audio mode MT. The tester will then prompt the test operator to press either the volume up or volume down key. The IUT should then transmit the Store Ringer Combination message. This test will also check the ranges of the ringer tones.

Already tested in §2_5.37.

2.6.7 Store Handsfree Volume (MT $08) received by the IUT.

The tester sends the IUT a define audio mode MT. The tester will then prompt the test operator to press either the volume up or volume down keys. This test will also check the ranges of the handsfree volume.

Already tested in §2_5.37.

2.6.8 Store Ringing Volume (MT $09) received by the IUT.

The tester sends the IUT a define audio mode MT. The tester will then prompt the test operator to press either the volume up or volume down keys. This test will also check the ranges of the ringer volume.

Already tested in §2_5.37.

2.6.9 Store Music Volume (MT $0A) received by the IUT.

The tester sends the IUT a define audio mode MT. The tester will then prompt the test operator to press either the volume up or volume down keys. This test will also check the ranges of the music volume.

Already tested in §2_5.37.

I. - Minet Commands

Table 1/MTS22-D is a list of all the MiNET commands. A gray box indicates that it is supported by the product indicated by the column.

Table 1/MTS22-D

PBX to Set Commands

Commands	ProductSS4001+/SS6001+	SS4100/SS6100	SS4200/SS6200	SS4300/SS6300	Console	ALM	DMP	PKM/SS622	AIM	SS40001	SS40015	SS40025/SS41125	SS41150	SS40900
(01) Send String														

(02) Send Individual Character														
(03) Send Individual Flashing Character														
(04) Position Cursor														
(05) Shift Display Field Left														
(06) Shift Display Field Right														
(07) Roll Display Field Left														
(08) Roll Display Field Right														
(09) Clear Display Lines														
(0A) Clear Field														
(0B) Flash Field ON														
(0C) Flash Field OFF														
(0E) Display Block Cursor														
(0F) Display Underscore Cursor														
(10) Blank Cursor														
(11) Copy Buffer Lines														
(12) Clear Screen														
(13) Delay Update of a Field														
(14) Display Pre-Stored Prompts														
(15) Store Prompt Text														
(16) Send Prompt														
(17) Deactivate Prompt														
(18) Display Buffer N														
(19) Set LCD Indicators														
(1A) Load LCD Flash Rates														
(1B) Send Signal														
(1C) Ringer Cadence High Continuous														
(1D) Ringer Cadence High Pulsed														
(1E) Ringer Cadence Low Continuous														
(1F) Ringer Cadence Low Pulsed														
(20) Send DTMF														
(22) Request Cursor Position														
(23) Start Precise Ringer Candencing														
(24) Set CODEC Gain														
(26) Start Voice Set Ringer														
(27) Define Initial Conditions														
(28) Define Audio Mode														
(29) Send Data Module Command														
(2A) Delete Character														
(2B) Cancel Delay Update of a Field														
(2C) Initialize Display														
(2D) Select Stored Prompt														

(2E) Define 5 x 7 Special Character														
(30) Define Character Pattern														
(33) Setup														
(34) Start New Call Tone										?	?	?	?	?
(35) Wireless														
(BF 0A) Start B Channel														
(C0) Request Set ID														
(C3) Read Set Memory														
(C5) Request Maintenance Data														
(C7) Request Set Reset														
(C8) Load Address														
(C9) Load Data														
(CA) Variable Length Load Data														
(CB) Checksum Request														
(CD) Run														
(CE) Write Set Memory														
(CF) Define Channel and MiLINK Parameter														
(D0) Configure Hardware														
(D2) Datagram										?	?	?	?	?
(D3) Local Programmable Key Control														
(D4) Miscellaneous Indicator Control														
(D5) Set Relay														
(D6) Pager Module Ringing														

I. - Minet Keycode Designations

Table 2/MTS22-D indicates the keycode designations for the devices indicated in the column. The ALM and the DMP do not have any keys and are thus not indicated in the table.

Figure 1/MTS22-D through Figure 6/MTS22-D illustrate the key numbers for the various devices. All values are in decimal.

Table 2/MTS22-D

Mitel Keycode Designations

Key Number (in decimal)	Product SS401+	SS410	SS420	SS430	Console	PKM	AIM
0	U-00 D-40	U-00 D-40	U-00 D-40	U-00 D-40		U-00 D-40	U-00 D-40
1	U-01 D-41	U-01 D-41	U-01 D-41	U-01 D-41	U-00 D-90	U-01 D-41	
2	U-02 D-42	U-02 D-42	U-02 D-42	U-02 D-42	U-01 D-91	U-02 D-42	
3	U-03 D-43	U-03 D-43	U-03 D-43	U-03 D-43	U-02 D-92	U-03 D-43	
4	U-04 D-44	U-04 D-44	U-04 D-44	U-04 D-44	U-03 D-93	U-04 D-44	
5	U-05 D-45	U-05 D-45	U-05 D-45	U-05 D-45	U-04 D-94	U-05 D-45	
6	U-06 D-46	U-06 D-46	U-06 D-46	U-06 D-46	U-05 D-95	U-06 D-46	
7			U-07 D-47	U-07 D-47	U-06 D-96	U-07 D-47	
8			U-08 D-48	U-08 D-48	U-07 D-97	U-08 D-48	
9			U-09 D-49	U-09 D-49	U-08 D-98	U-09 D-49	
10			U-0A D-4A	U-0A D-4A	U-09 D-99	U-0A D-4A	
11			U-0B D-4B	U-0B D-4B		U-0B D-4B	
12			U-0C D-4C	U-0C D-4C		U-0C D-4C	
13	U-0D D-4D	U-0D D-4D	U-0D D-4D	U-0D D-4D		U-0D D-4D	
14						U-0E U-4E	
15					81	U-0F U-4F	

16	U-10 D-50	U-10 D-50	U-10 D-50	U-10 D-50	82	U-10 D-50	
17	U-11 D-51	U-11 D-51	U-11 D-51	U-11 D-51	83	U-11 D-51	
18		U-12 D-52	U-12 D-52	U-12 D-52	U-CC D-8C	U-12 D-52	
19		U-13 D-53	U-13 D-53	U-13 D-53	84	U-13 D-53	
20		U-14 D-14	U-14 D-14		85	U-14 D-14	
21		U-15 D-55	U-15 D-55		86	U-15 D-55	
22	U-16 D-56	U-16 D-56	U-16 D-56	U-16 D-56	U-C0 D-80	U-16 D-56	U-16 D-56
23		U-17 D-57	U-17 D-57	U-17 D-57	87	U-17 D-57	
24	U-18 D-58	U-18 D-58	U-18 D-58	U-18 D-58	88	U-18 D-58	
25	19	19	19	19	89	U-19 D-59	19
26	1A	1A	1A	1A	U-CE D-8E	U-1A D-5A	1A
27	1B	1B	1B	1B	8A	U-1B D-5B	1B
28	1C	1C	1C	1C	8B	U-1C D-5C	1C
29	1D	1D	1D	1D	8C	U-1D D-5D	1D
30	1E	1E	1E	1E	U-CF D-8F	U-1E D-5E	1E
31	1F	1F	1F	1F	U-E7 D-A7		1F
32	20	20	20	20	U-E8 D-A8		20
33	21	21	21	21	U-E9 D-A9		21
34	22	22	22	22	U-EA D-EA		22
35	23	23	23	23	U-EB D-AB		23
36	24	24	24	24	U-EC D-AC		24
37			U-25 D-65	U-25 D-65	U-ED D-AD		
38			U-26 D-66	U-26 D-66	U-E0 D-A0		

39			U-27 D-67	U-27 D-67	U-E1 D-A1		
40				U-28 D-68	U-E2 D-A2		
41				U-29 D-69	U-E3 D-A3		
42				U-2A D-6A	U-E4 D-A4		
43					U-E5 D-A5		2B
44					U-E6 D-A6		2C
45							2D
46							2E

Figure 1/MTS22-D

Superset 401+ Key values

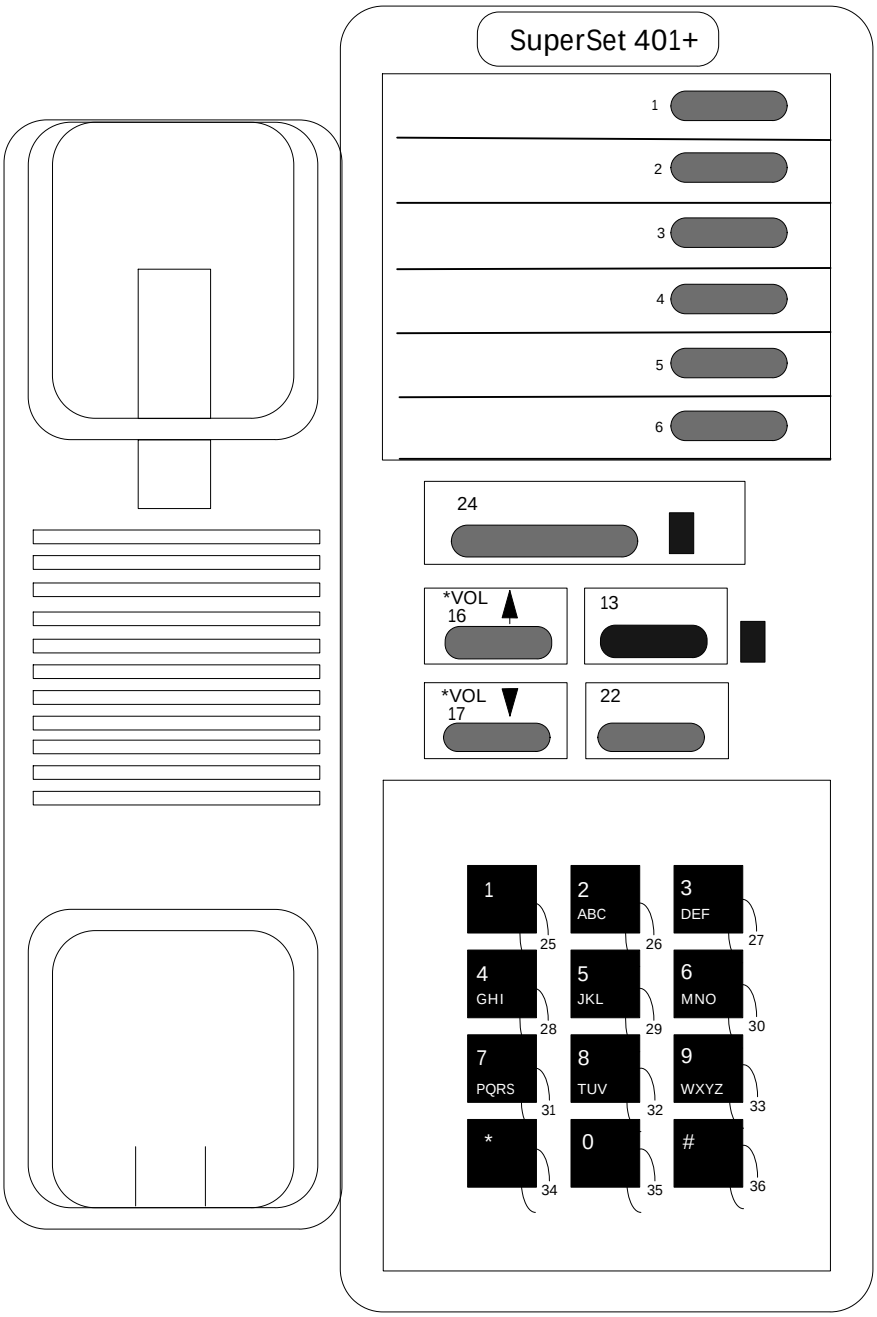


Figure 2/MTS22-D

Superset 410 Key values

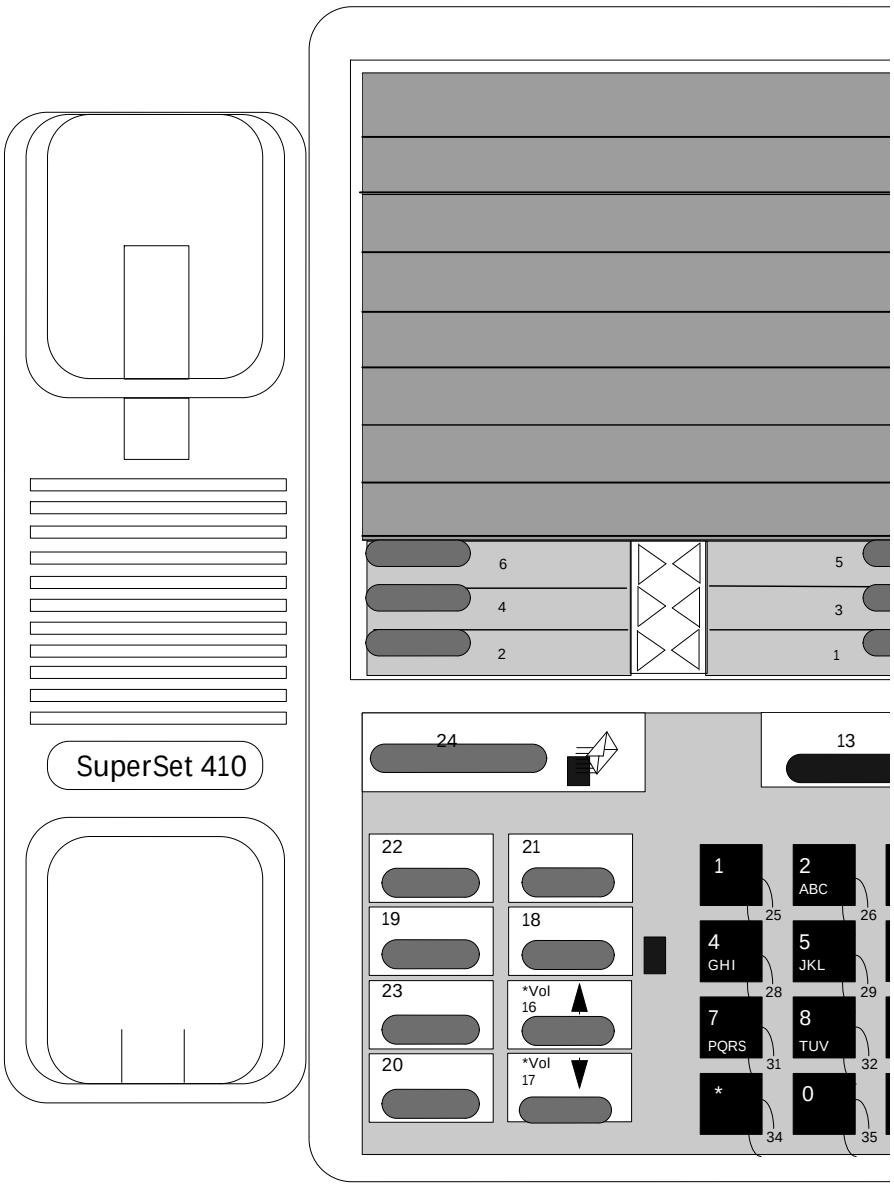


Figure 3/MTS22-D

Superset 420 Key values

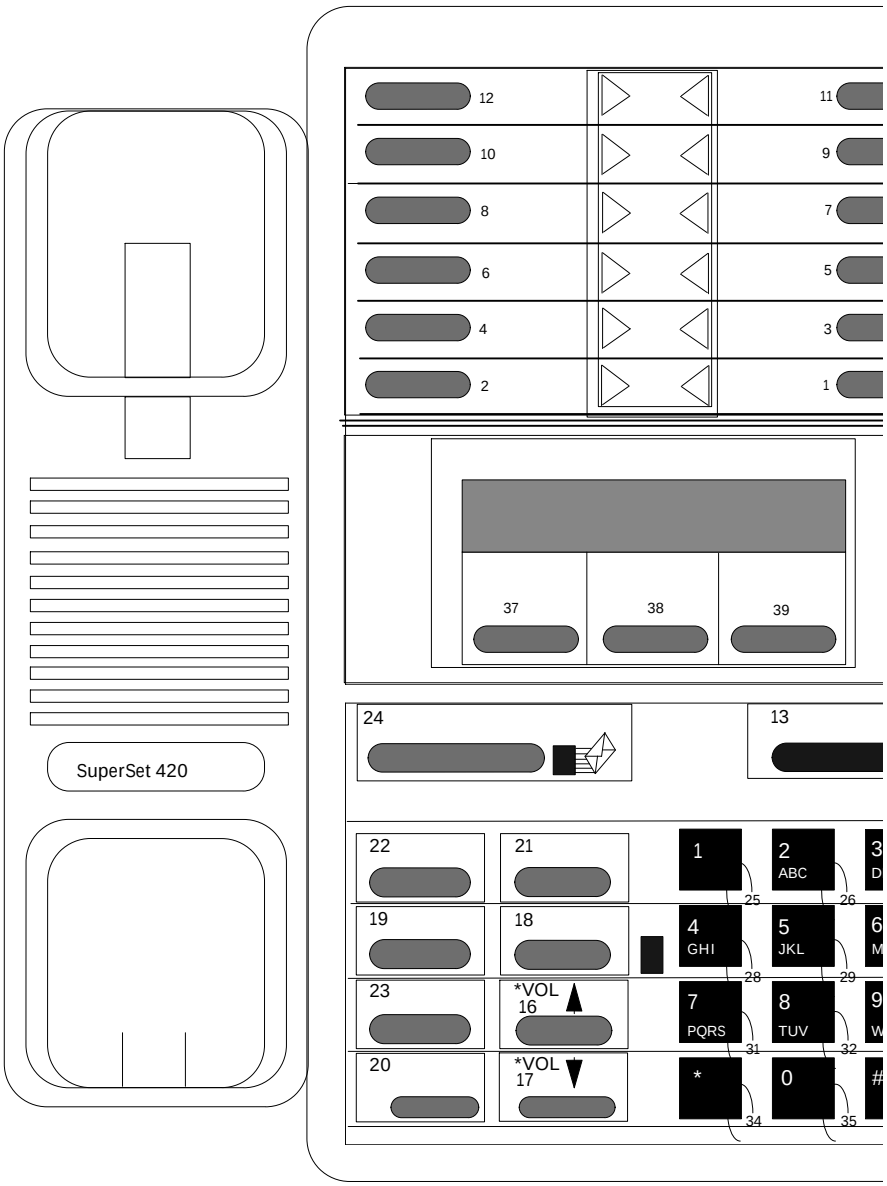


Figure 4/MTS22-D

Superset 430 Key values

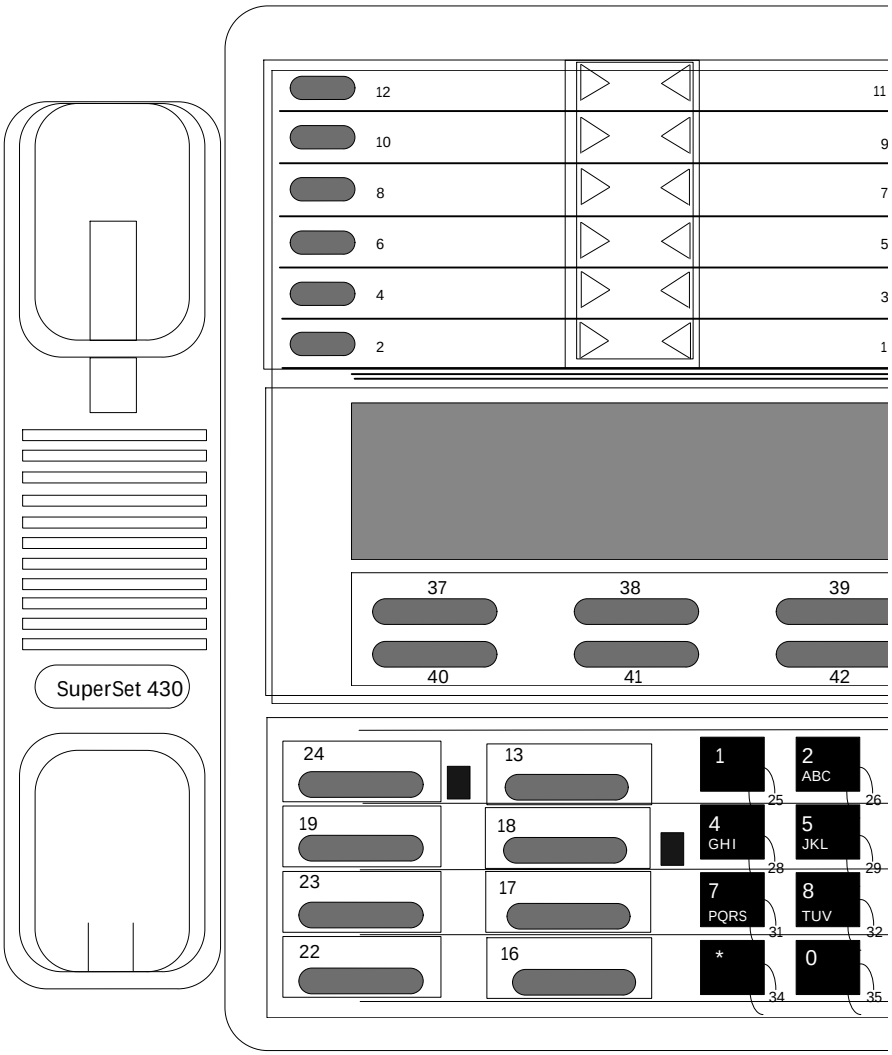
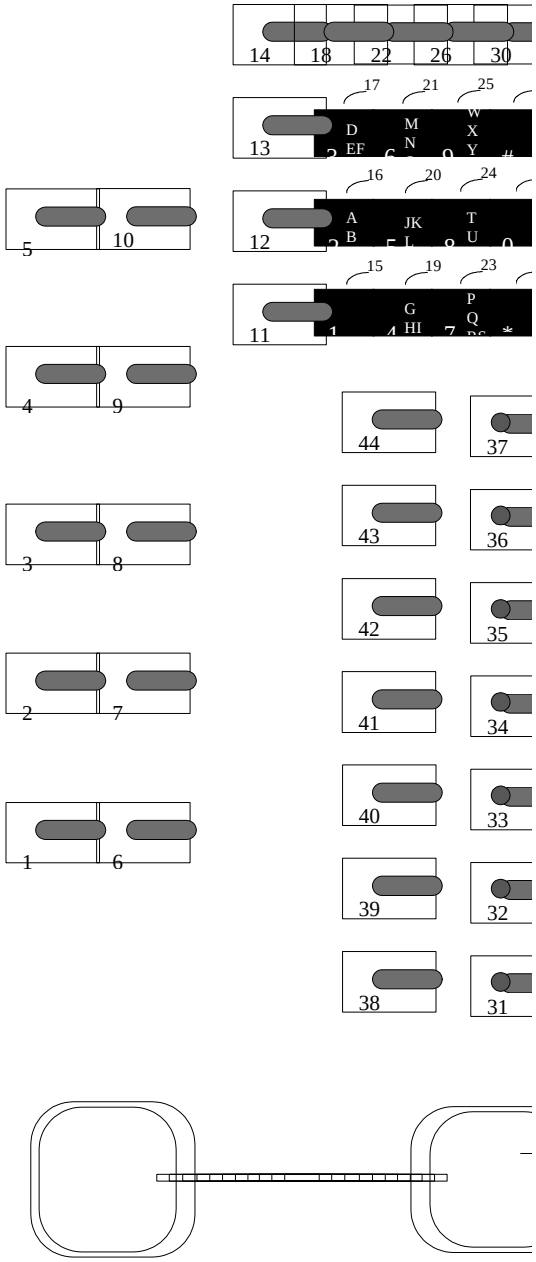


Figure 5/MTS22-D

Superset 6DN Key values



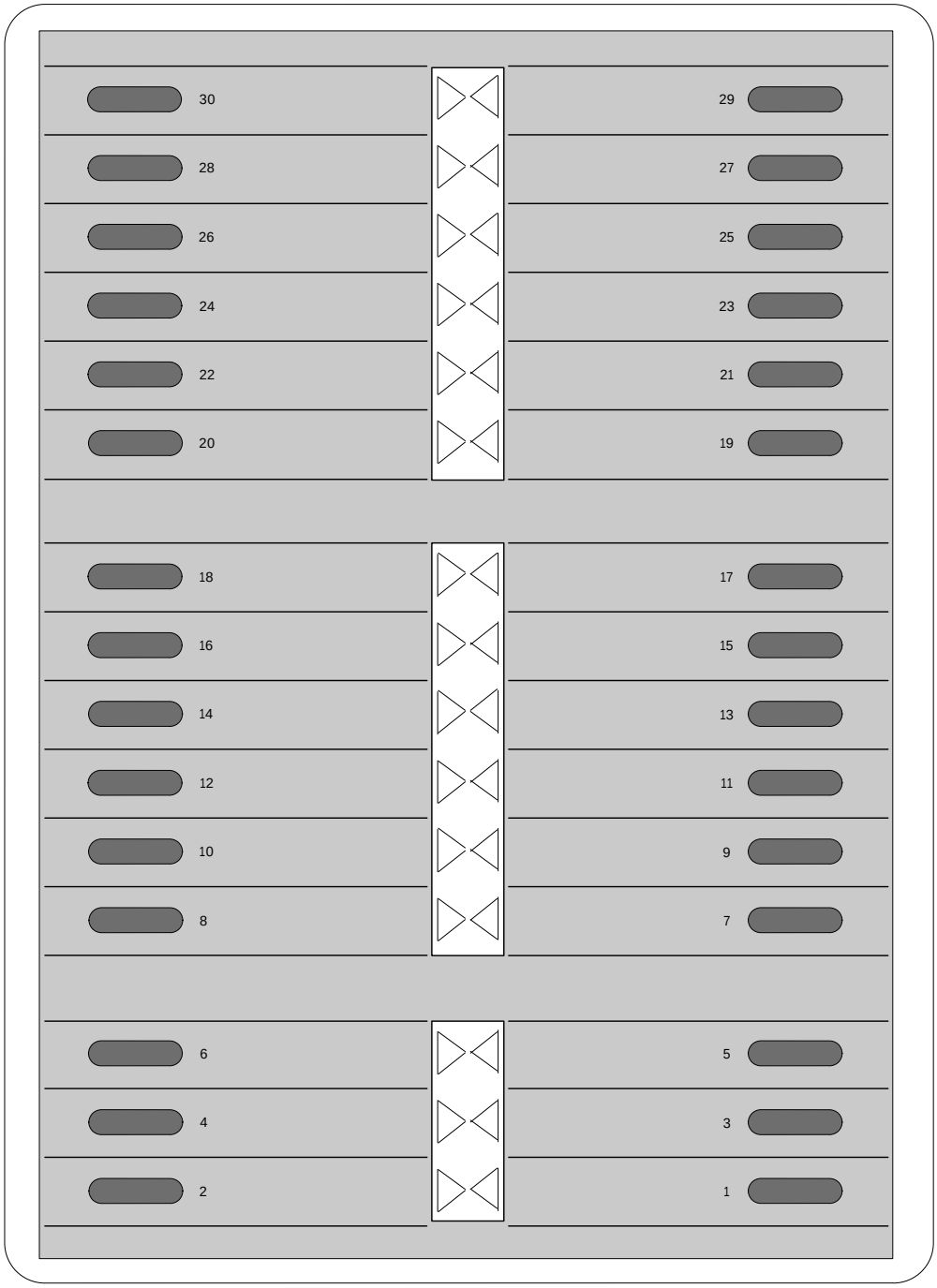


Figure 6/MTS22-D

Section E – Minet Extensions

1 Introduction

1.1 Purpose

This section describes the Minet IP extensions to MTS22. The message interface was written with consideration to the H323 Voice Gateway needs.

1.2 Acknowledgements

The Minet IP extensions are a culmination of the work that went into the USB phone, the E2T NIC, and the H323 Voice Gateway. Craig Frisch created the message templates used for the Minet IP extensions as described in this document, Andre Moskal provided valued input and leadership in the design of the message flows, but any inaccuracies or mistakes are solely the responsibility of the author, Chris Nason.

1.3 Change History

Change Level	Date	Author(s)	Comments
1W01	April 5, 2000	Chris Nason	First draft
1W02	July 4 2001	Chris Nason	Bring in Paul Provencal's Webset msgs
1W03	July 16 2001	Chris Nason	Revise main IPSP message structure to reflect new Webset messages
1W04	August 15 2001	Chris Nason	Webset messages no longer IPSP messages, change to MTS22
1W05	August 22 2001	Chris Nason	Added 'Modify TX Stream' and 'Modify RX Stream' messages
1W06	August 27 2001	Stephen Lee	Made part of MTS 22 document

2 Introduction

The Minet message collective is byte oriented and easily adaptable to Mitel's legacy PBX mix of assembly and higher level languages. The MinetIP extensions are structure based and are long word aligned. The implication of this is that a user with a packet Sniffer will see filler bytes in between short and long words.

2.1 Minet & Minet IP Messaging

In order to control a Mitel IP Phone, both Minet and Minet IP messages are required. A common message wrapper is defined to house the messages. The general message template consists of a Protocol Header and a Minet IP Message body that may or may not consist of an MTS22 Minet payload "wrapper".

Protocol Header:

ProtoType:	4 bytes , unsigned long integer, Protocol Type
devNum:	4 bytes , unsigned long integer, Device Number
msgType:	4 bytes , unsigned long integer, Message Type

Message body follows Protocol Header as shown in the structure below:

```
typedef struct _IPSP_MSG {
        PROTOCOL_HEADER_MSG hdr;
        union _msg {
                MINET_WRAPPER_MSG                MWM;
                DEVICE_REGISTRATION_MSG          DRM;
                DEVICE_REGISTRATION_ACK_MSG      DRAM;
                DEVICE_UNREGISTER_MSG            DUM;
                DEVICE_UNREGISTER_ACK_MSG        DUAM;
                OPEN_RX_STREAM_REQUEST_MSG       ORSRM;
                OPEN_RX_STREAM_ACK_MSG           ORSAM;
                CLOSE_RX_STREAM_REQUEST_MSG      CRSRM;
                CLOSE_RX_STREAM_ACK_MSG          CRSAM;
                OPEN_TX_STREAM_REQUEST_MSG       OTSRM;
                OPEN_TX_STREAM_ACK_MSG           OTSAM;
                CLOSE_TX_STREAM_REQUEST_MSG      CTSRM;
                CLOSE_TX_STREAM_ACK_MSG          CTSAM;
                APPLY_TONE_REQUEST_MSG           ATRM;
                REMOVE_TONE_REQUEST_MSG          RTRM;
                DEVICE_PING_REQUEST_MSG          DPRM;
                DEVICE_PING_ACK_MSG              DPAM;
                DEVICE_IP_UPDATE_REQUEST_MSG     DIURM;
                DEVICE_IP_UPDATE_ACK_MSG         DIUAM;
                MODIFY_RX_STREAM_REQUEST_MSG MRSRM;
                MODIFY_TX_STREAM_REQUEST_MSG MTSRM;
        } msg;
} IPSP_MSG;

typedef  struct {
        protocolType_t      protoType;
        deviceNumber_t      devNum;
        messageType_t       msgType;
 } PROTOCOL_HEADER_MSG;
```

Protocol Type:

INVALID_PROTOCOL_TYPE	0x00000000
MINET_MTS22	0x00000001
MITEL_INTERNAL	0x00000002

The Protocol Type denotes whether the message is a Minet IP message or an encapsulated Minet (MTS 22) message.

Device Number:

Phone	0x00000000
Device #1 ie PKM	0x00000001
Device #2	0x00000002
.....	.....
Device #n	0x0000000n

The Device Number denotes which entity sharing the same MAC address the messages are destined to or coming from.

Message Type:

INVALID_MESSAGE_TYPE	0x00000000
DEVICE_REGISTRATION	0x00000001
DEVICE_REGISTRATION_ACK	0x00000002
DEVICE_DEREGISTRATION	0x00000003
DEVICE_DEREGISTRATION_ACK	0x00000004
OPEN_RX_STREAM	0x00000005
OPEN_RX_STREAM_ACK	0x00000006
CLOSE_RX_STREAM	0x00000007
CLOSE_RX_STREAM_ACK	0x00000008
OPEN_TX_STREAM	0x00000009
OPEN_TX_STREAM_ACK	0x0000000a
CLOSE_TX_STREAM	0x0000000b
CLOSE_TX_STREAM_ACK	0x0000000c
MINET_WRAPPER	0x0000000d
APPLY_TONE	0x0000000e
REMOVE_TONE	0x0000000f
DEVICE_PING	0x00000010
DEVICE_PING_ACK	0x00000011
DEVICE_IP_UPDATE	0x00000012
DEVICE_IP_UPDATE_ACK	0x00000013
MODIFY_RX_STREAM	0x00000014
MODIFY_TX_STREAM	0x00000015
INVALID_MSG_TYPE	0x00000017

2.2 Minet IP Registration Sequence

When the IP Phone powers up or resets, it must register with the PBX. The phone originates a Registration Request and receives a Registration Acknowledgement in return. The system checks the Device ID of the phone (its MAC address) and verifies if it has it in the CDE database. If not, the system sends the phone an MTS22 Minet for PIN Request. The phone buffers the key entries and sends up one message containing the PIN Reply (also an MTS22 Minet message).

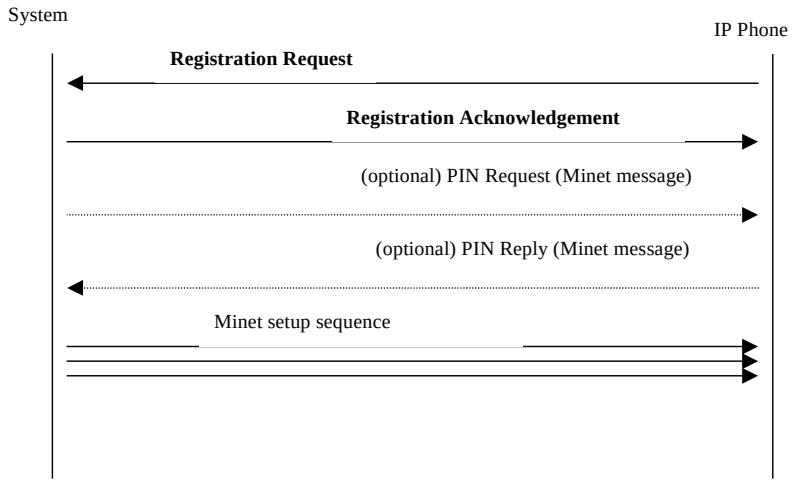


Figure 1: Registration sequence upon power up or reset.

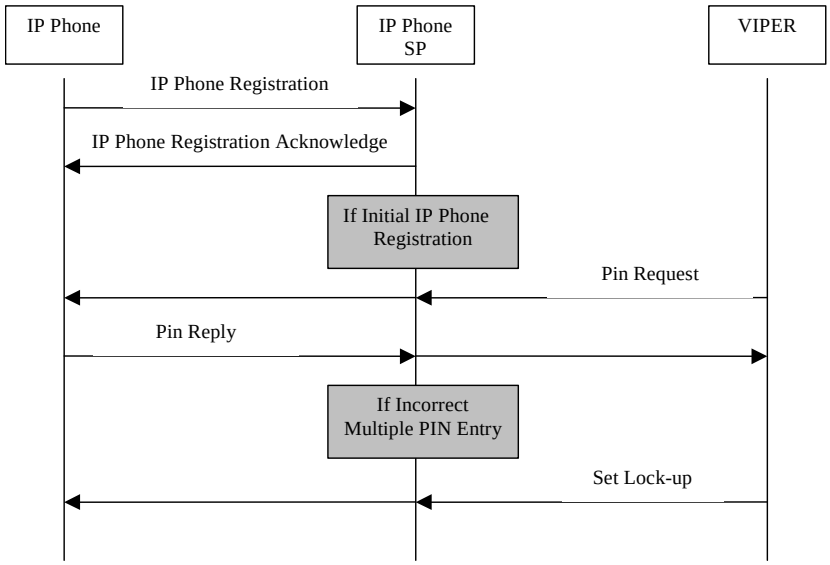


Figure 2: IP Phone registration with PIN

2.2.1 Device Registration request message sent from the IP Phone

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_REGISTRATION

DEVICE_REGISTRATION_MSG

devId: 6 unsigned byte array
- mac_addr[6] MAC address of Phone.

  Note that due to long word alignment, there may be 2 bytes of filler between the MAC address and the next defined field.

devType: 4 bytes , unsigned long integer, Type of device (i.e., SET, PKM, ...)

devNumber: 4 bytes , unsigned long integer, Number of device: Master, Slave01, Slave02, ...

ipAddress: structure
- ip_addr 4 bytes , unsigned long integer, IP Address of device,
- ip_port 2 bytes , unsigned short integer, port number of protocol medium.

  Note that due to long word alignment, there may be two bytes of filler between this field and the next.

DeviceCaps: structure: Functionality supported by this device
- strmCodec 4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Or'd into this field.
- numTxStreams: 4 bytes , unsigned long integer, Number of Tx streams supported by the device
- numRxStreams: 4 bytes , unsigned long integer, Number of Rx streams supported by the device
- prefStrmFrameSizeInMS: 4 bytes , unsigned long integer, Devices preferred frame size for streams (in ms)
- silenceSupp: 4 bytes , unsigned long integer:
  - silenceSupp=0: device does not support silence suppression
  - silenceSupp=1: device supports silence suppression
- toneGeneration: 4 bytes , unsigned long integer:
  - toneGeneration =0: device does not support local tone generation.
  - toneGeneration =1: device supports local tone generation

2.2.2 Device Registration request Acknowledgment message sent from system

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_REGISTRATION_ACK

DEVICE_REGISTRATION_ACK_MSG

reqStatus:	4 bytes , unsigned long integer, Success/Failure Result of the request
sysToken:	4 bytes , unsigned long integer, System defined "token" that must be passed back with any follow up message related to this message ie Device Unregister.

2.2.3 Device De-Registration Request message sent from IP Phone.

Device De-Registration Request message sent from IP Phone.
ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_DEREGISTRATION

Note that the IP Phone will not unregister itself, but rather an associated device such as a PKM may be removed and hence deregistered.

DEVICE_UNREGISTER_MSG

sysToken:	4 bytes , unsigned long integer, System defined "token" taken from the Registration Acknowledgment from the system.
devType:	4 bytes , unsigned long integer, Type of device (i.e., SET, PKM, etc...)
devNumber:	4 bytes , unsigned long integer, Number of device: Master, Slave01, Slave02, ...
ipAddress:	structure
ip_addr	4 bytes , unsigned long integer, IP Address of device,
ip_port	2 bytes , unsigned short integer, port number of protocol medium.

2.2.4 Device De-Registration Acknowledgment message sent from system

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_DEREGISTRATION_ACK

DEVICE_UNREGISTER_ACK_MSG

reqStatus:	4 bytes , unsigned long integer, Success/Failure Result of the request
devNumber:	4 bytes , unsigned long integer, Number of device: Master, Slave01, Slave02, ...

2.2.4.1 DETAILED DESCRIPTION OF REGISTRATION PARAMETERS

devType:

INVALID_DEVICE_TYPE 0x00000000

IP_SUPERSET4001	0x00000001
IP_SUPERSET4015	0x0000009f
IP_SUPERSET4025	0x000000a0
WEBSET	0x000000a4
PKM	0x0000009d
AIM	0x00000006
SYMBOL_PROXY	0x00000007
SYMBOL_SET	0x00000008
TELEWORKER_PROXY	0x00000009
TELEWORKER_SET	0x0000000a
E2T_PROXY	0x0000000b
DMP_PROXY	0x0000000c
DMP_MOH	0x0000000d
SC2000IP	0x000000a6
IP_APPLICATION	0x000000b0
WIRELESS_MINET	0x000000b1
IP_TRUNK_PORT	0x0000000e
MAX_DEVICE_TYPE	0x0000000f

devNumbers:

MASTER_DEVICE 0x00000000
Where Set=0, and any attached devices will be numbered MASTER_DEVICE + n where n >= 1

reqStatus (Success/failure codes):

MTL_SUCCESS	0x00000000
MTL_FAILURE	0x00000001
MTL_NO_PERMISSIONS	0x00000002
MTL_NO_RESOURCES	0x00000003
MTL_INVALID_DEVICE	0x00000004
MTL_INVALID_REQUEST	0x00000005

devCodecs bitmap:

NO_CODEC_SUPPORT	0x0	(000 00000000)
G711_ULAW64	0x1	(000 00000001)
G711_ALAW64	0x2	(000 00000010)
G728	0x4	(000 00000100)
G729	0x8	(000 00001000)
G729_ANNEXB	0x10	(000 00010000)
G729_ANNEXA_w_ANNEXB	0x20	(000 00100000)
G723	0x40	(000 01000000)
G7231_ANNEXC	0x80	(000 10000000)
Placeholder1	0x100	(001 00000000)
Placeholder2	0x200	(010 00000000)

Placeholder3	0x400 (100 00000000)
INVALID_CODEC	0x7FF (111 11111111)

2.3 Minet IP Device Remote ICMP Echo (PING) Message

The PING request to the phone is followed by a PING results message sent by the phone back to the system.

2.3.1 Device ICMP Echo (Ping) request to the phone

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_PING

DEVICE_PING_REQUEST_MSG

hostIpAddress:		structure
	ip_addr	4 bytes , unsigned long integer, IP Address of device to PING,
	ip_port	2 bytes , unsigned short integer, port number is IGNORED. Note that due to long word alignment, there may be two bytes of filler following this field.
numRequests		4 bytes , unsigned long integer, Number of ping requests to send
pktSize		4 bytes , unsigned long integer, Size of data packet to send (in bytes)
pktDelay		4 bytes , unsigned long integer, Inter packet delay in Milliseconds
timeOut		4 bytes , unsigned long integer, Ping request timeout in Milliseconds
qosLevel		4 bytes , unsigned long integer, no longer used. This field may be reused in the future to be TOS level instead of Layer 2 802.1p

2.3.2 Device ICMP Echo (Ping) results sent from the phone to the system

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_PING_ACK

DEVICE_PING_ACK_MSG

hostIpAddress:		structure
	ip_addr	4 bytes , unsigned long integer, IP Address of device that was PINGed,
	ip_port	2 bytes , unsigned short integer, port number is IGNORED. Note that due to long word alignment, there may be two bytes of filler following this field.

pktsSent	4 bytes , unsigned long integer, Number of ICMP echo requests sent
pktsRecv	4 bytes , unsigned long integer, Number of ICMP echo replys received
pktLoss	4 bytes , unsigned long integer, Percentage of packets lost
rttMax	4 bytes , unsigned long integer, Maximum round trip time (in milliseconds)
rttMin	4 bytes , unsigned long integer, Minimum round trip time (in milliseconds)
rttAvg	4 bytes , unsigned long integer, Average round trip time (in milliseconds)

2.3.2.1 Detailed Description of PING Parameters

QosLevel: NO LONGER USED!

QOS_LEVEL_NONE	0xffffffff
QOS_LEVEL_0	0x00000000
QOS_LEVEL_1	0x00000001
QOS_LEVEL_2	0x00000002
QOS_LEVEL_3	0x00000003
QOS_LEVEL_4	0x00000004
QOS_LEVEL_5	0x00000005
QOS_LEVEL_6	0x00000006
QOS_LEVEL_7	0x00000007

2.4 Minet IP Apply Tone Message

2.4.1 Apply Tone device tone generation request message to the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = APPLY_TONE

APPLY_TONE_REQUEST_MSG

sysToken:	4 bytes , unsigned long integer, System defined "token" that must be passed back with the Remove Tone request.
sysStrmID:	4 bytes , unsigned long integer, System provided stream ID which maps the voice streams to legacy B channels
tone[MAX_COMPLEX_TONE]:	array of tone structures of frequencies the DSP is to play
on_T1	2 bytes, unsigned long integer, Duration in ms of 1st ON period
off_T1	2 bytes, unsigned long integer, Duration in ms of 1st OFF period
on_T2	2 bytes, unsigned long integer, Duration in ms of 2nd ON period
off_T2	2 bytes, unsigned long integer, Duration in ms of 2nd OFF period
num_cycles	2 bytes, unsigned long integer, Number of times to repeat the ON/OFF sequence
tail	2 bytes, unsigned long integer, After num_cycles, 0 = leave tone off, 1 = on
freq_1	2 bytes, unsigned long integer, 1st frequency component in Hz
freq_2	2 bytes, unsigned long integer, 2nd frequency component in Hz
level_1	2 bytes, unsigned long integer, 1st frequency signal level
level_2	2 bytes, unsigned long integer, 2nd frequency signal level

action	2 bytes, unsigned long integer, indicates the action to take on completion of the tone. The actions are either to continue to the next tone descriptor, reconnect to the audio stream, or just stop. Note that due to long word alignment, there may be 2 bytes of filler following this field.
toneId:	4 bytes , unsigned long integer, System Tone ID of the tone being applied
inject;	4 bytes , unsigned long integer, specify whether to inject the tone on top of voice or not. This is unused by the phone since the tone will always take precedence over voice.

2.4.2 Remove Tone device tone generation request message to the phone

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = REMOVE_TONE

REMOVE_TONE_REQUEST_MSG

sysToken:	4 bytes , unsigned long integer, System defined "token" that was given with the Apply Tone request.
sysStrmID:	4 bytes , unsigned long integer, System provided stream ID which maps the voice streams to legacy B channels
tone[MAX_COMPLEX_TONE]:	array of tone structures of frequencies the DSP was playing out to the CODEC that it is to remove. Note that this is IGNORED BY IP PHONE
on_T1	2 bytes, unsigned long integer, Duration in ms of 1st ON period
off_T1	2 bytes, unsigned long integer, Duration in ms of 1st OFF period
on_T2	2 bytes, unsigned long integer, Duration in ms of 2nd ON period
off_T2	2 bytes, unsigned long integer, Duration in ms of 2nd OFF period
num_cycles	2 bytes, unsigned long integer, Number of times to repeat the ON/OFF sequence
tail	2 bytes, unsigned long integer, After num_cycles, 0 = leave tone off, 1 = on
freq_1	2 bytes, unsigned long integer, 1st frequency component in Hz
freq_2	2 bytes, unsigned long integer, 2nd frequency component in Hz
level_1	2 bytes, unsigned long integer, 1st frequency signal level
level_2	2 bytes, unsigned long integer, 2nd frequency signal level
action	2 bytes, unsigned long integer, indicates the action to take on completion of the tone. The actions are either to continue to the next tone descriptor, reconnect to the audio stream, or just stop.

2.4.2.1 Detailed Description of TONE Parameters

inject:

NOT_INJECTED	0x00000000
NORMAL_INJECTION	0x00000001
MAX_TONE_INJECT	0x00000002
MAX_COMPLEX_TONE	3

action:

NEXT	0x00000000
RECONNECT	0x00000001
STOP	0x00000002

2.5 Minet IP Connection Control Message

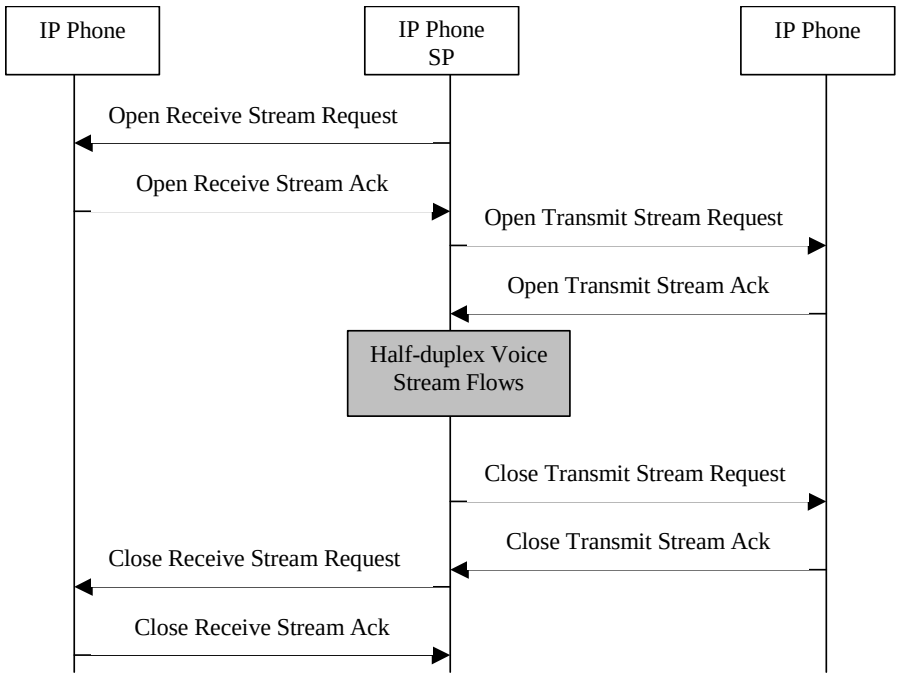


Figure 3: Connection Control message flows

2.5.1 Open Receive Stream Request to the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = OPEN_RX_STREAM

OPEN_RX_STREAM_REQUEST_MSG

sysToken:	4 bytes, unsigned long integer, System defined "token" that must be passed back with the corresponding Close Receive Stream Request .
sysStrmID:	4 bytes, unsigned long integer, System provided stream ID. This field denotes the B channel the connection should assume.

strmCodec		4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Or'd into this field.
strmFrameSizeInMS		4 bytes, unsigned long integer, Preferred CODEC frame size for the RX stream (in milliseconds)
isMulticast		4 bytes, unsigned long integer isMulticast =0: no Multicast, ignore mcIpAddress. isMulticast =1: the stream must be bound to the mcIpAddress Multicast address.
mcIpAddress:		structure
	ip_addr	4 bytes, unsigned long integer, Multicast address to receive on
	ip_port	2 bytes , unsigned short integer, Multicast port number to receive on. Note that due to long word alignment, there may be two bytes of filler following this field.
SrcIpAddress:		structure: IGNORED BY THE IP PHONE.
	ip_addr	4 bytes, unsigned long integer, The ip address of the device that will be transmitting to the phone.
	ip_port	2 bytes , unsigned short integer, port number used by the device that will be transmitting to the phone. Note that due to long word alignment, there may be two bytes of filler following this field.
noSilence		4 bytes, unsigned long integer, noSilence =0: no silence suppression applied by the transmitting end noSilence =1: silence suppression is being applied by the transmitting end

2.5.2 Open Receive Stream Acknowledgement from the IP Phone to the system:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = OPEN_RX_STREAM_ACK

OPEN_RX_STREAM_ACK_MSG

reqStatus:		4 bytes, unsigned long integer, Success/Failure Result of the request
sysToken:		4 bytes, unsigned long integer, System provided "token" from the request message
rxConnectionID:		4 bytes, unsigned long integer, Device selected stream/connection identifier. The IP Phone returns the value of the sysStrmID (B channel) in this field
rxStrmIpAddress:		structure
	ip_addr	4 bytes, unsigned long integer, The local ip address that will receive stream

ip_port	2 bytes , unsigned short integer, local port number to receive on.

2.5.3 Close Receive Stream Request from the system to the IP Phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = CLOSE_RX_STREAM

CLOSE_RX_STREAM_REQUEST_MSG

sysToken:	4 bytes, unsigned long integer, System defined "token" that was given with the Open Receive Stream Request .
sysStrmID:	4 bytes, unsigned long integer, Id of RX stream/connection (B channel) to close

2.5.4 Close Receive Stream Acknowledgement from the IP Phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = CLOSE_RX_STREAM_ACK

CLOSE_RX_STREAM_ACK_MSG

reqStatus:	4 bytes, unsigned long integer, Success/Failure Result of the request
sysToken:	4 bytes, unsigned long integer, System provided "token" from the request message
rxStrmStats:	structure: Stream statistics upon closure
Packets.recv	4 bytes, unsigned long integer, number of RTP packets received
Bytes.recv	4 bytes, unsigned long integer, number of voice octets received
Errors.rxStream	4 bytes, unsigned long integer, number of RTP errors received
Jitter.rxStream	4 bytes, unsigned long integer, estimate of average jitter over duration of call.
Duration.rxStream	4 bytes, unsigned long integer, duration of call in seconds
IpAddress.src:	structure
ip_addr	4 bytes, unsigned long integer, the local ip address
ip_port	2 bytes , unsigned short integer, the local port number.

2.5.5 Open Transmit Stream Request to the IP Phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = OPEN_TX_STREAM

OPEN_TX_STREAM_REQUEST_MSG

sysToken:	4 bytes, unsigned long integer, System defined "token" that must be passed back with the corresponding Close Transmit Stream Request .

sysStrmID:		4 bytes, unsigned long integer, System provided stream ID. This field denotes the B channel the connection should assume.
strmCodec		4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Or'd into this field.
strmFrameSizeInMS		4 bytes, unsigned long integer, Preferred CODEC frame size for the TX stream (in milliseconds)
destStrmIpAddress:		structure
	ip_addr	4 bytes, unsigned long integer, The IP address of the device to transmit to.
	ip_port	2 bytes , unsigned short integer, port number used by the device that will be transmitting to the phone. Note that due to long word alignment, there may be two bytes of filler following this field.
qosLevel		4 bytes, unsigned long integer, QoS level requested. If 0xffffffff, then no 802.1Q tag, else if 0-7, assume 802.1Q tag and set priority field to the qosLevel. **NO LONGER USED**
noSilence		4 bytes, unsigned long integer noSilence =0: disable silence suppression on the Tx stream noSilence =1: enable silence suppression on the Tx stream

2.5.6 Open Transmit Stream Acknowledgement from the IP Phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = OPEN_TX_STREAM_ACK

OPEN_TX_STREAM_ACK_MSG

reqStatus:		4 bytes, unsigned long integer, Success/Failure Result of the request
sysToken:		4 bytes, unsigned long integer, System provided "token" from the request message
txConnectionID:		4 bytes, unsigned long integer, Device selected stream/connection identifier. The IP Phone returns the value of the sysStrmID (B channel) in this field
txStrmIpAddress:		structure
	ip_addr	4 bytes, unsigned long integer, The local IP address that will transmit stream
	ip_port	2 bytes , unsigned short integer, local port number the phone will transmit from.

2.5.7 Close Transmit Stream Request to the IP Phone

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = CLOSE_TX_STREAM

CLOSE_TX_STREAM_REQUEST_MSG

sysToken:	4 bytes, unsigned long integer, System defined "token" that was given with the Open Transmit Stream Request .
sysStrmID:	4 bytes, unsigned long integer, Id of TX stream/connection (B channel) to close

2.5.8 Close Transmit Stream Acknowledgement from the IP Phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = CLOSE_TX_STREAM_ACK

CLOSE_TX_STREAM_ACK_MSG

reqStatus:	4 bytes, unsigned long integer, Success/Failure Result of the request
sysToken:	4 bytes, unsigned long integer, System provided "token" from the request message
txStrmStats:	structure: Stream statistics upon closure
Packets.sent	4 bytes, unsigned long integer, number of RTP packets sent
Bytes.sent	4 bytes, unsigned long integer, number of voice octets sent
Errors.txStream	4 bytes, unsigned long integer, number of RTP errors sent. IGNORE, NOT RELEVENT
Jitter.txStream	4 bytes, unsigned long integer, estimate of average jitter over duration of call. IGNORE, NOT RELEVENT
Duration.txStream	4 bytes, unsigned long integer, duration of call in seconds
IpAddress.dest:	structure
ip_addr	4 bytes, unsigned long integer, the local IP address used to Tx
ip_port	2 bytes , unsigned short integer, the local port number used to Tx.

2.5.8.1 Detailed Description of Connection Parameters

reqStatus (Success/failure codes):

MTL_SUCCESS	0x00000000
MTL_FAILURE	0x00000001
MTL_NO_PERMISSIONS	0x00000002
MTL_NO_RESOURCES	0x00000003
MTL_INVALID_DEVICE	0x00000004
MTL_INVALID_REQUEST	0x00000005

SysStrmID:
IP Set Stream IDs: (NOTE: TX is always even) used for sysStrmID of Tx & Rx connect requests

STREAM_ID_IP_SET_TX_1	0x00000000	// B1 TX
STREAM_ID_IP_SET_RX_1	0x00000001	// B1 RX
STREAM_ID_IP_SET_TX_2	0x00000002	// B2 TX
STREAM_ID_IP_SET_RX_2	0x00000003	// B2 RX

devCodecs bitmap:

NO_CODEC_SUPPORT	0x0	(000 00000000)
G711_ULAW64	0x1	(000 00000001)
G711_ALAW64	0x2	(000 00000010)
G728	0x4	(000 00000100)
G729	0x8	(000 00001000)
G729_ANNEXB	0x10	(000 00010000)
G729_ANNEXA_w_ANNEXB	0x20	(000 00100000)
G723	0x40	(000 01000000)
G7231_ANNEXC	0x80	(000 10000000)
Placeholder1	0x100	(001 00000000)
Placeholder2	0x200	(010 00000000)
Placeholder3	0x400	(100 00000000)
INVALID_CODEC	0x7FF	(111 11111111)

qosLevel:

QOS_LEVEL_NONE	0xffffffff
QOS_LEVEL_0	0x00000000
QOS_LEVEL_1	0x00000001
QOS_LEVEL_2	0x00000002
QOS_LEVEL_3	0x00000003
QOS_LEVEL_4	0x00000004
QOS_LEVEL_5	0x00000005
QOS_LEVEL_6	0x00000006
QOS_LEVEL_7	0x00000007

2.6 Minet IP Device IP Address Update Messages

2.6.1 Device IP address update request to the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_IP_UPDATE

DEVICE_IP_UPDATE_REQUEST_MSG

devNumber		4 bytes , unsigned long integer, Number of device: Master, Slave01, Slave02, ...
oldIpAddress:		structure
	ip_addr	4 bytes , unsigned long integer, old IP Address of device
	ip_port	2 bytes , unsigned short integer, old port number of device Note that due to long word alignment, there may be two bytes of filler following this field.
newIpAddress:		structure

ip_addr	4 bytes , unsigned long integer, new IP Address of device
ip_port	2 bytes , unsigned short integer, new port number of device

2.6.2 Device IP address update acknowledgement from the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = DEVICE_IP_UPDATE_ACK

DEVICE_IP_UPDATE_ACK_MSG

reqStatus:	4 bytes , unsigned long integer, Success/Failure Result of the request

2.6.2.1 Parameters Description

reqStatus (Success/failure codes):

MTL_SUCCESS	0x00000000
MTL_FAILURE	0x00000001
MTL_NO_PERMISSIONS	0x00000002
MTL_NO_RESOURCES	0x00000003
MTL_INVALID_DEVICE	0x00000004
MTL_INVALID_REQUEST	0x00000005

devNumbers:

MASTER_DEVICE 0x00000000
Where Set=0, and any attached devices will be numbered MASTER_DEVICE + n where n >= 1

2.6.3 Modify RX stream request to the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = MODIFY_RX_STREAM

sysToken:	4 bytes , unsigned long integer, System defined "token" that must be passed back with any follow up message related to this message.

MODIFY RX STREAM_REQUEST_MSG

sysToken	4 bytes , unsigned long integer, system defined token
sysStrmID:	4 bytes , unsigned long integer, system stream id
strmCodec	4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Or'd into this field.

2.6.4 Modify RX stream acknowledgement from the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = MODIFY_RX_STREAM_ACK

MODIFY RX STREAM_ACK_MSG

reqStatus: 4 bytes , unsigned long integer, Success/Failure Result of the request

2.6.4.1 Parameters Description

reqStatus (Success/failure codes):

MTL_SUCCESS	0x00000000
MTL_FAILURE	0x00000001
MTL_NO_PERMISSIONS	0x00000002
MTL_NO_RESOURCES	0x00000003
MTL_INVALID_DEVICE	0x00000004
MTL_INVALID_REQUEST	0x00000005

2.6.5 Modify TX stream request to the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = MODIFY_TX_STREAM

sysToken: 4 bytes , unsigned long integer, System defined "token" that must be passed back with any follow up message related to this message.

MODIFY TX STREAM_REQUEST_MSG

sysToken	4 bytes , unsigned long integer, system defined token
sysStrmID:	4 bytes , unsigned long integer, system stream id
strmCodec	4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Or'd into this field.

2.6.6 Modify TX stream acknowledgement from the phone:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,….n
msgType = MODIFY_TX_STREAM_ACK

MODIFY TX STREAM_ACK_MSG

reqStatus: 4 bytes , unsigned long integer, Success/Failure Result of the request

2.6.6.1 Parameters Description

reqStatus (Success/failure codes):

MTL_SUCCESS	0x00000000
MTL_FAILURE	0x00000001
MTL_NO_PERMISSIONS	0x00000002
MTL_NO_RESOURCES	0x00000003
MTL_INVALID_DEVICE	0x00000004
MTL_INVALID_REQUEST	0x00000005

APPROVALS

Approvals indicate acceptance of the document contents as a whole and its suitability for release and use at the change level indicated.

APPROVED BY:Keith Richardson SIGNATURE: _____

POSITION: AVP DATE SIGNED: _____

DEPARTMENT: Technical Requirements and Standards

APPROVED BY:Kirk Mandy SIGNATURE: _____

POSITION: Vice President DATE SIGNED: _____

DEPARTMENT: Research and Development

APPROVED BY:Ron Evans SIGNATURE: _____

POSITION: Vice President DATE SIGNED: _____

DEPARTMENT: Marketing and Product Management