ERICSSON

Mobile Communications

EDACS®
Data Gateway

CONFIGURATION REFERENCE MANUAL

TABLE OF CONTENTS

INTRODUCTION

GENERAL

This manual is a guide to configuring the EDACS® Data Gateway (EDG) for proper operation. In particular, it describes the command set and syntax for the *SYSTEM.TXT* file, which is a special file located in the /cnfg directory of the EDG hard drive. Examples of *SYSTEM.TXT* files for various system configurations are also included.

The *SYSTEM.TXT* commands have been designed to allow a quick and easy setup for users with a simple system configuration, yet provide the necessary power to users who are designing very complex systems. Because this release of the EDG incorporates the use of Internet Protocol (IP), a general knowledge of IP is assumed on the part of the reader, since many commands are directly related to configuration of system IP parameters.

USING THE EDG CONFIGURATION FILE

The EDG configures itself upon startup based on the contents of the *SYSTEM.TXT* configuration file. Without this file, the EDG will not boot. It is in ASCII format and can be created and modified with any ASCII based text editor. A generic *SYSTEM.TX*T file is provided on each system's hard drive, which may be used as a starting point for configuring your system.

GENERAL RULES

The following are some general rules about creating and editing the *SYSTEM.TXT* file for the EDG.

- The *SYSTEM.TXT* command parser is not case sensitive. Lower case characters are automatically read as upper case characters. However, one exception to this is any file names entered. The EDG hard drive's file system is UNIX based and, therefore, all file names are case sensitive.
- Unless specifically stated otherwise, any number of spaces or tab characters may be freely placed within the file as field separators.
- If a space or tab character is required in a command syntax, the command description explicitly states that.
- The EDG only recognizes tabs, line feeds, carriage returns, DOS end of file (CTRL-Z), and printable characters. Any other characters cause a syntax error.
- The fixed disk and floppy disk drives are referred to as "01.02" and "01.01", respectively, when using a full path name.

- All file and directory names on the hard disk follow a UNIX format, are case sensitive, must begin with an alphabetic character, and may contain up to 12 characters.

- All file and directory names on the floppy disk follow a DOS format. Names are in upper case, and may not exceed eight characters, plus a three character extension, separated by a period.

- When using a file's full path name, separate the drive name, directories, and file name with the forward slash character ("/"). An example is "01.02/subdir1/subdir2/FILE.EXT".

The *SYSTEM.TXT* file contains the statements that configure all the system parameters that allow the EDG to function properly. There are basically three types of statements that are entered in this file: commands, table entries, and headings Commands are used to configure a particular system parameter, table entries contain data that is used for various EDG internal tables, and headings identify groups of related commands or table entries.

♦ A heading statement can have one of two possible forms:
[heading]
[heading *field_value***]**

The **heading** value is a predefined text string containing no space characters. The brackets, '[' and ']', which identify that this is a heading statement, must be entered. Space characters may be entered before and after the opening and closing brackets, if desired.

For those heading statements that require it, *field_value* must be entered, which is a variable value within the range and in the format specified in the heading statement description. The **heading** and *field_value* must be separated by one or more space characters.

Heading statements are followed by a command block or a table. A command block is a group of command statements (described below) entered on lines in the file below the heading. A table is a group of table entries (described below) entered on lines in the file below the heading.
Any heading statement that is not applicable for a particular configuration may be omitted from the file, or may be entered in the file with no command block or table. There may be blank lines entered in a command block or table.

All heading statements, with their associated command blocks, may be placed in any order within the file.

- Each command statement has the following syntax:
  keyword *field_values*
  The **keyword** value is a predefined text string containing no space characters. The *field_values* are one or more values entered by the user, in the format and within any range specified in the command description. One or more space characters must separate **keyword** and the first field value. Multiple field values are also separated by one or more space characters.

  Each command statement must be entered on a separate line in the file and may only appear under its associated heading statement. Any command that is not related to the command block's heading will be ignored and will generate a warning. Commands may be placed in any order within the command block.

- Each table entry is composed of one or more fields, where a field is a value entered by the user in the format and within any range specified in the table description. Table entries may be placed in any order within the table, with one entry per line.

- The comment character, '#', may be inserted freely within the file. Any text to the right of the comment character is ignored by the parser. If a comment is to be on the same line as a heading or command statement, one or more space characters must precede the '#' character.

- All required numeric values may be entered in decimal or hexadecimal format. A hexadecimal format is identified by appending the character 'h' or 'H' to the value. The hex characters 'a' through 'f' may be entered in upper or lower case. For example, all of the following represent the same value:
  1ab9h, 1ab9H, 1Ab9H, 1AB9h

- Some fields require an IP address value. This address must consist of four separate numeric values, with each value separated by a period, and no space characters. An address may be entered in decimal or hexadecimal format, with the hexadecimal format as described previously. If using decimal format, the range for each of the four values is 0 - 255, while the range for each of the values using a hexadecimal format is 00 - FF. Note that leading zeros may be used for each of the four values, or may be omitted. For example, all of the following represent the same address:
  143.001.04.013, 143.1.004.13, 8F.01.4.dh, 8f.01.04.0dH

- Some fields allow a numeric range. A range consists of either a single numeric value, or two numeric values separated by a '-' character. Space characters are permitted before and after the '-'. The first value must be numerically less than the second value.

- Some fields allow a numeric list. A list is composed of one or more numeric values, each separated by a comma. Space characters are permitted before and after each comma. The values may be entered in the list in any order.

This manual uses the following conventions for configuration file command descriptions.

Convention	Usage
italics	A field in italics indicates a variable that you must replace with a value.
BOLD	A word in boldface letters indicates a key word that must be used. You may enter the keyword in upper or lower case letters.
OPTION	Items shown in non-boldface uppercase characters are a specific option that can be selected. You can use them in upper or lower case in the file.
A|B|C	When options appear with a vertical bar separator, you must choose between one of the options.

On the following page is a summary of the available SYSTEM.TXT commands and headings. Each heading is followed by either its command block, containing all available commands for that heading, or its table entry format. Detailed descriptions and examples for each command and heading follow the summary page.

COMMAND SET SUMMARY

[Board *board_number***]**
HDI_Port_Hosts *port_number host_list*
HDI_Port_Parity ODD | EVEN | NONE
Label "*board_label*"
Load *loadfile*
Port_Direction *port_number* IN | OUT | BI | OFF
TSI_Baud 19200 | 9600
TSI_EDACS_ID *unit_id*
TSI_Hosts *host_list*
Type CAP | TSI_Master | TSI_Slave *master_board_number* | HDI

[IP]
CAP_Default_Gateway *ip_address*
CAP_Ext_Address *ip_address*
Host_Name *host_name*
Int_Board_Address *board_number ip_address*
Int_Network_ID *ip_network_id*
Int_Routing_Address *ip_address*

[System]
Max_Msgs *msg_count*
Max_TELNET_Sessions *session_count*
Max_FTP_Sessions *session_count*
Msg_Timeout *timeout_value*
Outbound_Msg_Delay *delay_value*
Transport_Layer_Protocol *protocol_type_code*
TSI_Anti_Biasing TRUE | FALSE

[IMC *imc_number***]**
Label *label*
EDACS_IP_Network_ID *network_address*

[IP_Map_ID_Table]
id_type id_range start_ip_address
.
.
id_type id_range start_ip_address

[IP_CAP_Ext_Routing_Table]
HOST | NET *destination next_gateway*
.
.
destination next_gateway

[Network_Layer_RDT_Table]
id_type id_range
.
.
id_type id_range

COMMAND DESCRIPTIONS

[Board] Command Heading

Format: [Board *board_number*]

The commands under this heading define a board and its associated parameters within the EDG. Each board is assigned a unique identification number which is entered in the *board_number* field. Board numbers must be contiguous, starting at the number "1", which must be always be assigned to the CAP board.

[Board] HDI_PORT_HOSTS Command

Syntax: **HDI_Port_Hosts** *port_number host_list*

Requirement: Valid for HDI type boards only. This command is required for each active port (i.e. a port for which the Port_Direction command was stated with a direction of IN, OUT, or BI).

Default Value: None.

Description: This command is used to assign EDACS host ids to a given HDI port.

port_number must be a value in the range 0 - 3, corresponding to port 0 through port 3 on the board.

host_list contains one or more EDACS host ids, with each id separated by a comma. A host id is any value in the range 1 - 63. Data messages destined for a given RDI host will only be routed through those HDI output ports with the destination host id in its host list. Any messages from an RDI host received through an HDI port will have its source host id defined as the lowest valued host id assigned to that port.

Comments: A host id may be assigned to more than one port on the same HDI or multiple HDIs. If a host id is assigned to two or more HDIs, output messages will be rotored to each of the HDIs. At the individual HDI level, output messages are also rotored through all output ports to which the destination host id is assigned.

Every host id assigned to an HDI must be assigned to a master TSI. This may be done explicitly using the **TSI_Hosts** command (which allows you to designate to which TSI each host id is to be assigned), or it will be done by default by the EDG during startup.

Every host id assigned to an HDI port must be mapped to an IP address under the **[IP_Map_ID_Table]** heading. This may be done explicitly, or it will be done by default by the EDG during startup. In the default case, the IP address assigned to the host id will be 200.0.0.XX, where XX is the host id.

[Board] HDI_PORT_HOSTS Command: *Continued*

Example: There are two HDIs in the system, with four ports active on the first HDI, and only port 2 active on the second HDI. Port 0 on HDI #1 will process messages for host ids 1 - 5, ports 1 & 2 on HDI #1 will process messages for host id 10, and port 3 on HDI #1 and port 2 on HDI #2 will process messages for host id 2. The following commands would be entered for HDI #1:

HDI_Port_Hosts 0 1, 2, 3, 4, 5
HDI_Port_Hosts 1 10
HDI_Port_Hosts 2 10
HDI_Port_Hosts 3 2

The following command would be entered for HDI #2:

HDI_Port_Hosts 2 2

If no IP addresses were assigned explicitly to the host ids in the [IP_Map_ID_Table], the following addresses would be assigned by default during startup:

Host ID 1: 200.0.0.1
Host ID 2: 200.0.0.2
Host ID 3: 200.0.0.3
Host ID 4: 200.0.0.4
Host ID 5: 200.0.0.5
Host ID 10: 200.0.0.10

[Board] HDI_PORT_PARITY Command

Syntax: **HDI_Port_Parity** ODD | EVEN | NONE

Requirement: Valid for HDI type boards only. This is an optional command.

Default Value: Defaults to EVEN parity.

Description: This command is used to indicate the type of parity to be used on all ports of a single HDI for serial communications.

The type of parity must be one of the three values specified above, either ODD, EVEN, or NONE.

This command applies to all ports of the HDI board for which it is specified. However, each HDI board in the system may be defined with its own parity type.

Example: There are two HDIs in the system. The first HDI is to use even parity on all its ports. The second HDI is to use no parity on all its ports. Since even parity is the default, the HDI_Port_Parity command does not have to be specified for the first HDI. However, the following command would need to be entered for the second HDI:

HDI_Port_Parity NONE

[Board] LABEL Command

Syntax: **Label** "*label*"

Requirement: Optional for all boards

Default: A text string consisting of the board number and type will be assigned, e.g. "BD 2 TSI Mas", "BD 1 CAP".

Description: This allows a user defined text string of up to 12 case sensitive characters to be assigned to the board. It is only referenced in messages to the terminal and in any EDG generated reports.

label may contain any printable ASCII character, including space characters, and must be enclosed by double quotation marks. Labels need not be unique. As many characters as will fit on a single line may be entered as the *label,* but only the first twelve will be read and saved.

Example: Board number 2 is a TSI Master type. No Label command is entered for it. Its default label will be "BD 2 TSI Mas".

Example: It is desired to assign the label "Ralph the TSI wonder dog" to the current board. The following command would then be entered:

Label "Ralph the TSI wonder dog"

However, since only the first 12 characters are saved, the actual label assigned to this board would be "Ralph the TS".

[Board] LOAD Command

Syntax: **Load** *filename*

Requirement: Required for all boards

Default: None

Description: This command tells the system loader which application file to load onto the board.

filename should be the full pathname of the file to be loaded, and should not be enclosed in quotation marks. The full pathname should include the device volume number, the directory path, and the file name. Note that all names are case sensitive.

Example: The hard drive volume number is 01.02. The floppy drive volume number is 01.01.

The EDG loader automatically copies any application files found on a floppy disk to the /loads directory on the EDG hard drive, with the filename in all capital letters. Therefore, to load the CAP application code in this case, the following command would be entered:

Load 01.02/loads/CAP.SX

[Board] PORT_DIRECTION Command

Syntax: **Port_Direction** *port_number* IN | OUT | BI | OFF

Requirement: Valid for TSI_MASTER, TSI_SLAVE, and HDI type boards only. At least one port on each board must be assigned a non-OFF direction, i.e. a direction of IN, OUT, or BI.

Default: An OFF direction is assigned to any port for which this command is not explicitly stated.

Description: This command defines the direction of message flows for a given port on the board.

port_number must be a value in the range 0 - 3, corresponding to port 0 through port 3 on the board.

The direction must be one of the four possible text strings as shown above:

- **IN** - Only input data to the EDG is accepted through this port; no output data will be routed through it even if no other ports are available.
- **OUT** - Output data may be sent through this port. However, it does not prevent the EDG from accepting data received through the port.
- **BI -** Input and output messages may be routed freely through the port.
- **OFF -** A port that is not being used should be set to this value, although any port that is not explicitly defined a direction will be set to an OFF status.

Example: Ports 1 and 2 of the current board are to be active, with port 1 to only receive messages, while port 2 may transmit and receive messages. The following commands would be entered:

Port_Direction 1 IN
Port_Direction 2 BI
Port_Direction 0 OFF
Port_Direction 3 OFF

Note that the 3rd and 4th commands are not necessary, since a port that is not assigned a direction will be set to OFF.

[Board] TSI_BAUD Command

Syntax: **TSI_Baud** 9600 | 19200

Requirement: Valid for TSI Master type boards only. It is an optional command.

Default: The default is a 19200 baud rate.

Description: This command sets the baud rate at which the master TSI communicates with the **D**ata **I**nterface **M**odule (**DIM**) located in the EDACS IMC/CEC.

Example: There are 2 master TSIs in the system. The first one communicates with the IMC/CEC at 19200 baud. The second one communicates at 9600 baud. Since the default baud rate is 19200, no command is required for the first TSI to set its baud rate. However, the following command would be entered under the [Board] heading for TSI #2:

TSI_Baud 9600

[Board] TSI_EDACS_ID Command

Syntax: **TSI_EDACS_ID** *unit_id*

Requirement: Required for all TSI Master type boards only. Not valid for other board types.

Default: None.

Description: Assigns an EDACS Logical Unit ID to a TSI Master, which is required by the IMC. The unit id assigned to a TSI Master must be unique among all other unit ids assigned to all other entities within the EDACS system, including other TSI Masters, radios, consoles, etc.

unit_id is a value in the range 64 - 16382. Normally, allowable EDACS unit ids are in the range 1 - 16382. However, the EDG requires that a TSI may not be assigned one of the possible host IDs in the range 1 - 63.

Comments: The EDACS System Manager does not currently support an EDG TSI as a valid type when entering a unit id in the database. However, to avoid potential EDACS system problems, any unit id assigned to a TSI Master should be entered in the System Manager database to avoid inadvertent assignment to another entity. This could be done by adding it as a radio type, but assigning a label such as "EDG TSI" for reference.

Example: There are 2 master TSIs in the system. TSI #1 is assigned unit id 600, while TSI #2 is assigned unit id 601.

The following command would be entered under the [Board] heading for TSI #1:

TSI_EDACS_ID 600

The following command would be entered under the [Board] heading for TSI #2:

TSI_EDACS_ID 601

[Board] TSI_HOSTS Command

Syntax: **TSI_Hosts** *host_list*

Requirement: Valid for TSI Master type boards only. It is an optional command.

Default: When network layer RDTs have been defined (i.e. there are one or more entries entered under the **[Network_Layer_RDT_Table]** heading), host ids 1 - 15 are automatically assigned to all master TSIs in round robin fashion. Also, any host ids found in the **[IP_Map_ID_Table]** that have not been assigned to a master TSI by the user, will be assigned in round robin fashion as well, whether network layer RDTS are defined or not.

Description: This command is used to assign EDACS host ids to all active ports of a TSI Master board. The IMC receives this information and will send messages destined for a given host id only to the TSI Master to which the host id is assigned.

host_list contains one or more EDACS host ids, with each id separated by a comma. A host id is any value in the range 1 - 63.

The same host id may not be assigned to more than one master TSI.

If network layer RDTs are not being used (i.e. there are no entries under the **[Network_Layer_RDT_Table]** heading), the following must also be done for each and every host id assigned to a TSI:

- it must be mapped to an IP address under the **[IP_Map_ID_Table]** heading.
- it must be assigned to an HDI port, using the **HDI_Port_Hosts** command,
  or its IP address must be entered as the *destination_address* in an **[IP_CAP_Ext_Routing_Table]** HOST table entry,
  or the network portion of its IP address must match the *destination_address* of an **[IP_CAP_Ext_Routing_Table]** NET table entry,
  or the network portion of its IP address must match the IP network ID assigned to the CAP external network, via the **[IP] CAP_Ext_Address** command,
  or the network portion of its IP address must match the IP network ID assigned internally to the EDG, via the **[IP] Int_Network_ID** command.
  or the network portion of its IP address must match the IP network ID assigned to the EDACS network, via the **[IMC] EDACS_IP_Network_ID** command.

[Board] TSI_HOSTS Command: *Continued*

Example: Non-network layer RDTs are in use and there are two TSI masters in the system. TSI #1 will process messages destined for host ids 1-4, while TSI #2 will process messages destined for host ids 5-9.

The following command would be entered for TSI #1:

TSI_Hosts 1,2,3,4

The following command would be entered for TSI #2

TSI_Hosts 5,6,7,8,9

[Board] TYPE Command

Syntax: **Type** CAP | TSI_SLAVE *master_board_num* | TSI_MASTER | HDI

Requirement: Required for all boards

Default: None

Description: This command defines the board's type. One of three possible text strings shown above must be entered.

master_board_num is required only for a TSI_SLAVE board type. Every TSI slave board in the EDG must be linked to a board defined as a TSI master. Therefore, the board number of the master to which the slave is linked must be entered in this field.

Example: The current board is to be the CAP board. The following command would then be entered:

Type CAP

Example: The current board is to be a TSI Slave type. Board number 2 in the system is a TSI Master type, and is to be this board's master. The following command would then be entered:

Type TSI_SLAVE 2

[IP] Command Heading

Format: [IP]

This heading's command block contains commands that are used in setting up IP specific values within the EDG. Some commands are board specific, while others affect the overall EDG. Almost all **[IP]** commands described here are optional, with system defaults normally used in all but the more complex system configurations. See the command descriptions below and the examples given for more details.

[IP] INT_NETWORK_ID Command

Syntax:	**Int_Network_Id** *ip_network_id*
Requirement:	Optional command.
Default:	199.0.0.0
Description:	Assigns an IP network ID to the EDG internal network. All devices that communicate directly on the same network have the same unique network ID as part of their IP address. In this case, the network is the EDG backplane, and the devices are the CAP, TSI, and HDI boards. *ip_network_id* is a 32-bit value, in an IP address format, that obeys the rules for IP network ID values.
Example:	A class C network ID, 201.0.0.0, is to be assigned to the EDG internal network. The following command would then be entered under the [IP] heading: **Int_Network_Id** 201.0.0.0

[IP] INT_BOARD_ADDRESS Command

Syntax: **Int_Board_Address** *board_number ip_address*

Requirement: Optional command.

Default: The network id portion of a board's address is set to the **Int_Network_Id** value, while the host id portion is set to the board's number.

Description: Defines the internal IP address to be assigned to each board in the EDG.

board_number is the number of the board to which the address is to be assigned.

ip_address is the 32-bit address value, in an IP address format, to be assigned to the board.

Comments: All internal IP addresses assigned to boards must contain the same internal network ID value. The host ID portion of the address must be within the range of allowable values as determined by the network class:
If class A: (first octet of network id is in range 0 - 127), the host id portion of the address occupies the lower 24 bits.
If class B: (first octet of network id is in range 128 - 191), the host id portion of the address occupies the lower 16 bits.
If class C: (first octet of network id is in range 192 - 223), the host id portion of the address occupies the lower 8 bits.
No other network classes are allowed within the EDG.
Note that the host id bits may not be set to all ones or all zeros.

Example:

Internal_Network_ID	192.16.4.0	
Internal_Board_Address	1	192.16.4.1
Internal_Board_Address	2	192.16.4.8

In this example, the internal network ID defines a class C network, which means that the upper 24 bits of every internal board address must be the same value, namely 192.16.4. Since only the lower 8 bits are to be used for host IDs, the addresses that may be assigned to EDG boards may only be in the range 192.16.4.1 to 192.16.4.254. Addresses 192.16.4.0 and 192.16.4.255 may not be assigned to a specific device since to do so would violate IP addressing rules. Here, board 1 is assigned the address 192.16.4.1, while board 2 is assigned the address 192.16.4.8.

[IP] INT_BOARD_ADDRESS Command: *Continued*

Example: All the addresses in the class C network, 197.0.0.0, are available for use within the internal EDG network. There are 3 boards in the system. In this case, the internal board addresses may be used, with only the network id specified under the **[IP]** heading:

Int_Network_Id 197.0.0.0

The following addresses would then be assigned automatically:
Board 1: 197.0.0.1
Board 2: 197.0.0.2
Board 3: 197.0.0.3

[IP] INT_ROUTING_ADDRESS Command

Syntax: **Int_Routing_Address** *ip_address*

Requirement: Optional command. The only time this command might need to specified is when the EDG internal IP addresses are being set manually using the **Int_Board_Address** command.

Default: The network id portion of the address is set to the **Int_Network_Id** value, while the host id portion is set to 1 greater than the highest board number in the system.

Description: This is a unique IP address that is required by the EDG in order to perform proper internal IP routing.

ip_address is a unique 32-bit address value, in an IP address format. It must contain the EDG internal network ID, and the host ID portion of the address must be within the range of allowable values as defined by the internal subnet mask value.

Example: The internal network ID is to be 210.3.2.0, which is a class C network. Only addresses 210.3.2.20 through 210.3.2.23 are available for use. There are 2 boards in the system. The following commands would be entered under the **[IP]** section:

Int_Network_ID 210.3.2.0
Int_Board_Address 1 210.3.2.20
Int_Board_Address 2 210.3.2.21
Int_Routing_Address 210.3.2.22

Example: There are 3 boards in the system. The internal network ID is to be 210.3.2.0, which is a class C network. All addresses in the network are available for use. The following command is entered under the **[IP]** section:

Int_Network_ID 210.3.2.0

Board 1 will be assigned a default IP address of 210.3.2.1, board 2 will be assigned 210.3.2.2, and board 3 will be assigned 210.3.2.3. Since the internal routing address is not specified, and board 3 is the highest board number in the system, the default internal routing address will be 210.3.2.4.

[IP] CAP_DEFAULT_GATEWAY Command

Syntax: **CAP_Default_Gateway** *ip_address*

Requirement: Optional command.

Default: None

Description: Defines the default gateway address to which packets will be sent if all of the following conditions are met:

- The packet's destination address is not on the CAP's external network.
- The packet's destination address is not on the EDG internal network.
- The packet's destination address is not in the CAP's routing table.

ip_address is the 32-bit address value, in an IP address format, to be assigned as the default gateway. This address must exist on either the CAP's external network, or on the EDG internal network.

Comments: If a default gateway is not specified, any packet that meets all the conditions listed in the description above will be discarded.

Example: The CAP's external network ID is 10.0.0.0. Any packet received by the CAP with an unknown destination is to be sent to the address 10.34.78.6 on the CAP's external network. The command entered under the **[IP]** heading would be:

CAP_Default_Gateway 10.34.78.6

[IP] CAP_EXT_ADDRESS Command

Syntax: **CAP_Ext_Address** *ip_address*

Requirement: Required command.

Default: None

Description: Assigns an IP address to the external (i.e. foreplane) port of the CAP. While this command must always be specified, its value is significant only if the EDG is to be connected to an external IP network.

ip_address is the 32-bit address value, in an IP address format, to be assigned as the external CAP address. The network ID portion of *ip_address* must match the ID of the network to which it is connected, while the host ID portion of *ip_address* must be a unique number on that network.

Comment: The CAP external network ID is also derived from this value, based on the network class of the address:
If class A: (first octet of network id is in range 0 - 127), the network id is the first octet value, followed by zeros in the remaining three octets.
If class B: (first octet of network id is in range 128 - 191), the network id is the first and second octet values, followed by zeros in the remaining two octets.
If class C: (first octet of network id is in range 192 - 223), the network id is the first, second, and third octet values, followed by a zero in the fourth octet.
No other network classes are allowed within the EDG.

Example: **CAP_Ext_Address** 129.039.12.6

This command specifies that the external address assigned to the CAP is 129.039.12.6. Since this is a class B address, the CAP external network ID will be set to 129.39.0.0.

[IP] HOST_NAME Command

Syntax: **Host_Name** *name_string*

Requirement: Optional command.

Default: EDG

Description: Assigns an IP host name to the EDG system. Many network services, such as NFS, require a host name. This will be used for future product capability.

name_string is a case sensitive string of 1 to 31 alphanumeric characters, with no embedded spaces. The string should not be enclosed in quotation marks.

Example: "Division_8_EDG_System_1" is to be assigned as the IP host name of the EDG. The following command would be entered under the **[IP]** heading:

Host_Name Division_8_EDG_System_1

[System] Command Heading

Format: [System]

This heading's command block contains commands that are used in setting up global system parameters.

[System] MAX_FTP_SESSIONS Command

Syntax: **Max_FTP_Sessions** *session_count*

Requirement: Optional command.

Default: 2

Description: This command is used to specify the maximum number of FTP sessions that may be simultaneously active in the EDG from remote hosts.

session_count is a value in the range 0 to 4. A value of 0 indicates that no remote hosts may access the EDG via FTP.

Example: **Max_FTP_Sessions** 1

This command indicates that only one remote host may access the EDG via FTP. If a second host tries initiate an FTP session, it will be denied.

[System] MAX_MSGS Command

Syntax: **Max_Msgs** *msg_count*

Requirement: Optional command.

Default: 100

Description: This command is used to specify the maximum number of messages that may be queued for output from a master TSI or HDI. This is on a per board basis.

msg_count is the number of messages in the range 10 to 1000.

Comments: When a message is discarded due to this maximum message count being exceeded, an ICMP "Source Quench" message is sent back to the message source.

This command is used to fine tune the EDG system performance (also see the **Msg_Timeout** command). If the majority of messages being received are of small byte counts, and are arriving at a high rate, then this value should be set to the upper end of the scale in order to maintain a high average message throughput. However, if very large messages are being received along with many small messages, then this value should not be set too high. This will prevent the situation where a small number of large messages are using up all available memory on the board. If all memory is in use, any incoming messages will be discarded, no matter what the maximum count value is, resulting in a degradation of average message throughput.

Example: **Max_Msgs** 200

This command indicates that there may be a maximum of 200 messages queued for transmission at each individual master TSI or HDI. If there were 2 master TSIs and 1 HDI in the system, then 200 messages could be queued at each master TSI and 200 at the HDI, resulting in 600 messages that could be possibly be queued in the EDG at any instant. However, if the maximum count is reached on an individual board, any further messages received by that board will be discarded as long as its queue remains full, even if other boards of the same type are not filled to capacity. Messages received by other boards with non-full queues will not be affected, though.

[System] MAX_TELNET_SESSIONS Command

Syntax: **Max_TELNET_Sessions** *session_count*

Requirement: Optional command.

Default: 2

Description: This command is used to specify the maximum number of TELNET sessions that may be simultaneously active in the EDG from remote hosts.

session_count is a value in the range 0 to 4. A value of 0 indicates that no remote hosts may access the EDG via TELNET.

Example: **Max_Telnet_Sessions** 3

This command indicates that only 3 remote hosts may access the EDG via TELNET at the same time. If a fourth host tries initiate a TELNET session, it will be denied.

[System] MSG_TIMEOUT Command

Syntax: **Msg_Timeout** *timeout_value*

Requirement: Optional command.

Default: 30 seconds

Description: This command is used to specify the maximum amount of time, in seconds, that a message may be active in the EDG (either on an HDI or TSI) before being flushed and any outstanding message fragments discarded.

The timeout period begins when the first fragment of the message is received at the EDG.

timeout_value is the number of seconds in the range 15 to 1000.

Comments: When a message is discarded due to the timeout being exceeded, an ICMP "Fragment reassembly time exceeded" message is sent back to the message source.

If a transport layer is handling retransmissions, the following guidelines should be used to set the transport layer timeout value:

With RDI hosts not using ACK2 and all IP hosts, set it to a minimum of *timeout_value* plus five seconds.

With RDI hosts using ACK2, set it to a minimum of *timeout_value* plus 15 seconds.

With RDI hosts using a 45 second timeout for the ACK2 return (as defined in the RDI specification), *timeout_value* must be set to 30 seconds.

This command is used to fine tune the EDG system performance (also see the **Max_Msgs** command). If very large messages are being discarded by the EDG, the timeout value may need to be increased. However a timeout value that is too large may cause congestion due to interference with flow control by an optional upper (transport) layer, or due to external delays of incoming message fragments.

[System] MSG_TIMEOUT Command: *Continued*

Comments: Note the "time to live" value in the IP header is not used to timeout a message since this is a hop count, rather than an actual time value.

Example: **Msg_Timeout** 40

This command indicates that a timeout value of 40 seconds is to be applied separately to each message destined for output from a TSI or HDI. If the full message has not been transmitted 40 seconds after receiving the first fragment of the message, any and all remaining fragments will be discarded.

[System] OUTBOUND_MSG_DELAY Command

Syntax: **Outbound_Msg_Delay** *delay_value*

Requirement: Optional command.

Default: 300 milliseconds

Description: This command is used to specify the minimum delay to be introduced by the EDG between consecutive data calls to the same destination radio on the IMC network. The delay is computed as the time between the end of one data call and the beginning of the next data call.

delay_value is the number of milliseconds in the range 0 to 15000, and in increments of 10, i.e. 0, 10, 20, 30, ... 14990, 15000.

Comments: This forced delay is required due to the call setup and data processing time requirements at the receiving radio and RDI. Without such a throttling mechanism at the EDG, data loss would increase, due to two factors:

- Some radio models require a minimum amount of setup time between the end of one call and the beginning of the next call.
- Even after the radio has transmitted all the data to the RDI and is ready to accept the next call, the RDI may still be transmitting the data to the RDT.

The amount of delay to be specified here is dependent upon the particular radio model being used, and whether simulation of a full duplex link is desired between the EDG and any given RDT.

For one way outbound data traffic, i.e. data being sent out from the EDG, as of the writing of this document, the following values were successfully used for the radio model indicated:

FMD	100 (for unit data call sizes < 350 bytes)
	200 (for unit data call sizes > 350 bytes)
	600 (for group data calls)
MTD	200 (for unit data calls)
	600 (for group data calls)
Orion	0 (for unit data calls)
	100 (for group data calls)

[System] OUTBOUND_MSG_DELAY Command: *Continued*

The delay value can also be used to simulate a full duplex link, i.e. two way simultaneous traffic, between the EDG and the RDT. This is accomplished by making the delay long enough for the RDT to respond to a received message before the next message is sent from the EDG, as well as allowing the message response to be received by the EDG before the number of transmit retries on the next outbound message have been exhausted.

If throughput is not an issue, a good value to use is 7500 msec plus the amount of time it takes to process the message at the RDT. If throughput is an issue, the delay could be reduced to as low as 1000 msec. However, this may increase the outbound data error rate since the EDG may try to transmit the next outbound message before the inbound message response is received in full. Although the EDG will retry a message transmission three times, at two second intervals, before finally giving up and discarding the message, it is not desirable to effectively "waste" the available retries due to a delay value that is too small.

The delay value is more of a work around to a possible data congestion problem, rather than a solution. The best way to avoid congestion is to have the application software on the host or RDT slow down its message output.

It should be further noted that, in the case where the same radio is receiving both group and unit data calls consecutively, this delay factor will not help prevent the loss of data. This is due to the fact that the group call will have a destination id or address that is different from that of the unit call, and the EDG does not track which radios are assigned to which groups.

Example: **Outbound_Msg_Delay** 100

This command indicates that for two or more consecutive data calls with the same destination id or address, the TSI will wait a minimum of 100 milliseconds after the end of of one call before it attempts the next call to the same radio.

[System] TRANSPORT_LAYER_PROTOCOL Command

Syntax: **Transport_Layer_Protocol** *protocol_type_code*

Requirement: Optional command. Its value is referenced only when non-network layer RDTs are being used.

Default: 17, which is the protocol type code for UDP IP packets.

Description: When non-network layer RDTs are used, the message arrives at the TSI, from the RDT, with no IP header. However, all messages that pass through the EDG must have an IP header, which the TSI will create in this situation. One of the fields in this header is the protocol type code field. The value defined by this command is the value placed in that field by the TSI.

protocol_type_code is a value in the range 0 to 255. There are many standard protocol type codes defined, including the following:

1 ICMP
6 TCP
8 EGP
17 UDP

Example: **Transport_Layer_Protocol** 6

This command indicates that the TCP protocol type code will be placed in the IP header created by the TSI when non-network layer RDTs are in use.

[System] TSI_ANTI_BIASING Command

Syntax: **TSI_Anti_Biasing** TRUE | FALSE

Requirement: Optional command.

Default: FALSE.

Description: Indicates whether the BREN algorithm should be used for all data messages between all EDG TSIs and the RDTs communicating with them. This algorithm resolves the biasing problem that may result from certain message bit sequences. Note that if this command is specified TRUE, all RDIs in communication with the EDACS network must also be utilizing the BREN algorithm.

Example: **TSI_Anti_Biasing** TRUE

This command indicates that all TSIs will use the BREN algorithm to compose its messages, and that all RDIs communicating with the EDG will be required to use this algorithm for successful communications.

[IMC] Command Heading

Format: [IMC *imc_number*]

This heading is used to define an EDACS IMC/CEC (abbreviated as IMC) in the system. There must always be at least one IMC defined.

Every master TSI in the EDG is connected to an IMC. An IMC is defined using the **[IMC** *imc_number***]** statement, where *imc_number* is a value in the range 1 - 15. Currently, support exists only for the definition of a single IMC connected to an EDG, therefore, although a value must be placed in *imc_number*, it does not have any functional significance at this time.

[IMC] EDACS_IP_NETWORK_ID Command

Syntax: **EDACS_IP_Network_Id** *ip_network_id*

Requirement: Optional command.

Default: 140.XX.0.0, where XX is the IMC number defined in the [IMC] heading.

Description: Assigns an IP network ID to the current EDACS IMC being defined.

Each IMC/CEC has associated with it a single IP network ID, with all EDACS unit and group IDs assigned unique IP addresses within that network. The actual mapping of these IDs to IP addresses is defined in the table under the **[IP_Map_ID_Table]** heading.

It is important to note that each IP address assigned to an EDACS unit or group ID on a single IMC/CEC must have the same network ID portion, but a unique host ID portion. This is important since the EDG will use this premise to determine, in a multiple IMC/CEC configuration, which units and groups are located on which IMC/CEC, so that internal routing tables may be correctly built.

ip_network_id is a 32-bit value, in an IP address format, that obeys the rules for IP network ID values.

[IMC] EDACS_IP_NETWORK_ID Command: *Continued*

Comments: The class of the network id defines the available IP addresses that may be assigned to units and groups on the EDACS network for this IMC. :
If class A: (first octet of network id is in range 0 - 127), the host id portion of the address occupies the lower 24 bits.
If class B: (first octet of network id is in range 128 - 191), the host id portion of the address occupies the lower 16 bits.
If class C: (first octet of network id is in range 192 - 223), the host id portion of the address occupies the lower 8 bits.
No other network classes are allowed within the EDG.
Note that the host id bits may not be set to all ones or all zeros.

Example: A class B network ID, 150.0.0.0, is to be assigned to IMC 1. The following command would then be entered under the **[IMC 1]** heading:

EDACS_IP_Network_Id 150.0.0.0

The available addresses would then be 150.0.0.1 through 150.0.255.254 (address 150.0.0.0 would be the "all zeros" host id case, while 150.0.255.254 would be the "all ones" host id case, both of which are illegal).

Example: An IMC has been defined with the number 3, i.e. **[IMC 3]**. No IP network ID has been explicitly specified for this IMC, so the default value assigned will be 140.3.0.0.

[IMC] LABEL Command

Syntax: **Label** "*label*"

Requirement: Optional for all IMCs.

Default: A text string consisting of "IMC" and its number,, e.g. "IMC 1".

Description: This allows a user defined text string of up to 12 case sensitive characters to be assigned to the IMC. It is only referenced in messages to the terminal and in any EDG generated reports.

label may contain any printable ASCII character, including space characters, and must be enclosed by double quotation marks. Labels need not be unique. As many characters as will fit on a single line may be entered as the *label*, but only the first twelve will be read and saved.

Example: IMC 2 is defined, but no Label command is entered for it. Its default label will be "IMC 2".

Example: It is desired to assign the label "IMC KEY MOUSE" to IMC 3. The following command would then be entered under the **[IMC 3]** heading:

Label "IMC KEY MOUSE"

However, since only the first 12 characters are saved, the actual label assigned to this IMC would be "IMC KEY MOUS".

[IP_Map_ID_Table] Table Heading

Format: [IP_Map_ID_Table]

This heading precedes a table of entries that will map various types of EDACS IDs to IP addresses. The EDG requires a unique IP address to be assigned to every radio, host computer, and group ID involved in data communications, so this is not an optional table. This is necessary since every network data message sent through the EDG must have a source IP address, identifying the message sender, and a destination IP address, identifying the desired recipient. The source address allows the recipient to respond back to the sender, while the destination address is used to provide the correct routing through the EDG and any external IP networks.

There are no commands associated with this heading, only one or more single line entries. The entry format is described below.

[IP_Map_ID_Table] Entry

Entry Syntax: *id_type id_range start_ip_address*

Description: Each entry in this table maps either a single EDACS ID or a block of EDACS IDs to a single IP address or block of IP addresses, respectively. There is no limit to the number of entries that may be placed in this table, i.e. every possible EDACS ID may be assigned to a separate entry. Each entry consists of three required fields.

id_type consists of a single character identifying the type of EDACS ID being mapped. Valid values include:

- **u** - EDACS Unit ID.
- **g** - EDACS Group ID.
- **h** - EDACS Host ID. Although a host ID is actually a unit ID in the range 1 - 63, it must be distinguished, within the EDG, from a unit ID assigned to a radio. This is necessary for building of internal routing tables, and other error checking algorithms.

id_range is a single ID or range of IDs to be mapped to a single IP address or range of IP addresses, respectively. A range is entered in the form:

start_id - end_id

Unit ids may be in the range 16 - 16382, and group ids may be in the range 0 - 2047.

[IP_Map_ID_Table] Table Entry: *Continued*

start_ip_address is a single value in IP address format. If the preceding id_range field is a single value, then start_ip_address is the IP address assigned to the ID value. If, instead, id_range is a range of values, then start_ip_address identifies the beginning of a contiguous block of addresses to be assigned to the IDs.

Example: The following IP mapping must be performed:

- Host IDs 1 - 10 are to be assigned a block of IP addresses beginning at address 200.0.0.1.
- Unit ID 1000 is to be assigned address 130.2.0.1.
- Unit IDs 5000 - 9999 are to be assigned a block of IP addresses beginning at 130.2.1.0.
- All EDACS group IDS (0 - 2047) are to be assigned a block of IP addresses beginning at 130.2.128.0.

The table would then appear as follows:

[IP_Map_ID_Table]

h	1-10	200.0.0.1	# Host IDs
u	1000	130.2.0.1	# Special unit
u	5000-9999	130.2.1.0	# All other units
g	0-2047	130.2.128.0	# All possible group ids

[IP_CAP_Ext_Routing_Table] Table Heading

Format: [IP_CAP_Ext_Routing_Table]

This heading precedes a table of entries that make up the external routing table for the CAP board. Each entry in the table defines the correct routing for any messages destined for an external host that is not on the CAP's local external network, i.e. the host's network ID is not the same as the CAP external network ID.

There are no commands associated with this heading, only one or more single line entries. These entries are concatenated to the CAP's internal routing table, which is built automatically at startup. The entry format is described below.

NOTE:

The total number of entries in the CAP's routing table, which consists of the external entries defined here, plus the number of CAP internal routing table entries, plus any CAP default gateway entry, may not exceed 1024.

The number of CAP internal routing table entries will be equal to the number of unique host IDs assigned to HDIs, plus the number of IMCs connected to the EDG, plus three (one each for the EDG backplane Network Interface (NI), the foreplane NI, and the loopback address 127.0.0.1).

[IP_CAP_Ext_Routing_Table] Entry

Entry Syntax: HOST | NET *destination_address next_gateway*

Description: The *destination_address* type is specified using one of the two values shown above. If it refers to an individual destination, the HOST keyword should be specified. If it refers to a network, the NET keyword should be specified.

destination_address is the IP address of an external host, or an external network, that is not on the CAP's local external network, but can be reached via a device, i.e. a gateway, that is on the CAP's local external network.

next_gateway is the IP address of a host that is on the CAP's local external network through which messages may be routed to reach a destination on another network. Note that any next_gateway address must contain the same network ID as that of the CAP external address.

[IP_CAP_Ext_Routing_Table] Entry: *Continued*

Example: The CAP's external network ID is 10.0.0.0. A gateway with address 10.4.3.2 exists on this network through which the network 144.6.0.0 may be reached. Another gateway with address 10.20.2.79 exists on this network through which the single host device 6.1.1.16 may be reached. The table would then appear as follows:

[IP_CAP_Ext_Routing_Table]

NET	144.6.0.0	10.4.3.2
HOST	6.1.1.16	10.20.2.79

[Network_Layer_RDT_Table] Table Heading

Format: [Network_Layer_RDT_Table]

This heading precedes a table of entries that will indicate which EDACS unit and group IDs will be communicating with the EDG via RDTs that use a network layer header. This header allows the sender to specify its destination with an IP address, rather than an EDACS id.

There are no commands associated with this heading, only one or more single line entries. The entry format is described below.

[Network_Layer_RDT_Table] Entry

Entry Syntax: *id_type id_range*

Description: Each entry in this table specifies either a single EDACS ID or a block of EDACS IDs that will be sending and receiving messages using a network layer header. There is no limit to the number of entries that may be placed in this table, i.e. every possible EDACS ID may be assigned to a separate entry. Each entry consists of two required fields.

id_type consists of a single character identifying the type of EDACS ID being specified. Valid values include:

- **u** - EDACS Unit ID.
- **g** - EDACS Group ID.

id_range is a single ID or range of IDs. A range is entered in the form:

start_id - end_id

Unit ids may be in the range 16 - 16382, while group ids may be in the range 0 - 2047.

Example: The following EDACS ids will be associated with network layer RDTs:

- Unit IDs 300 and 1000 - 2000
- Group IDs 0 - 999

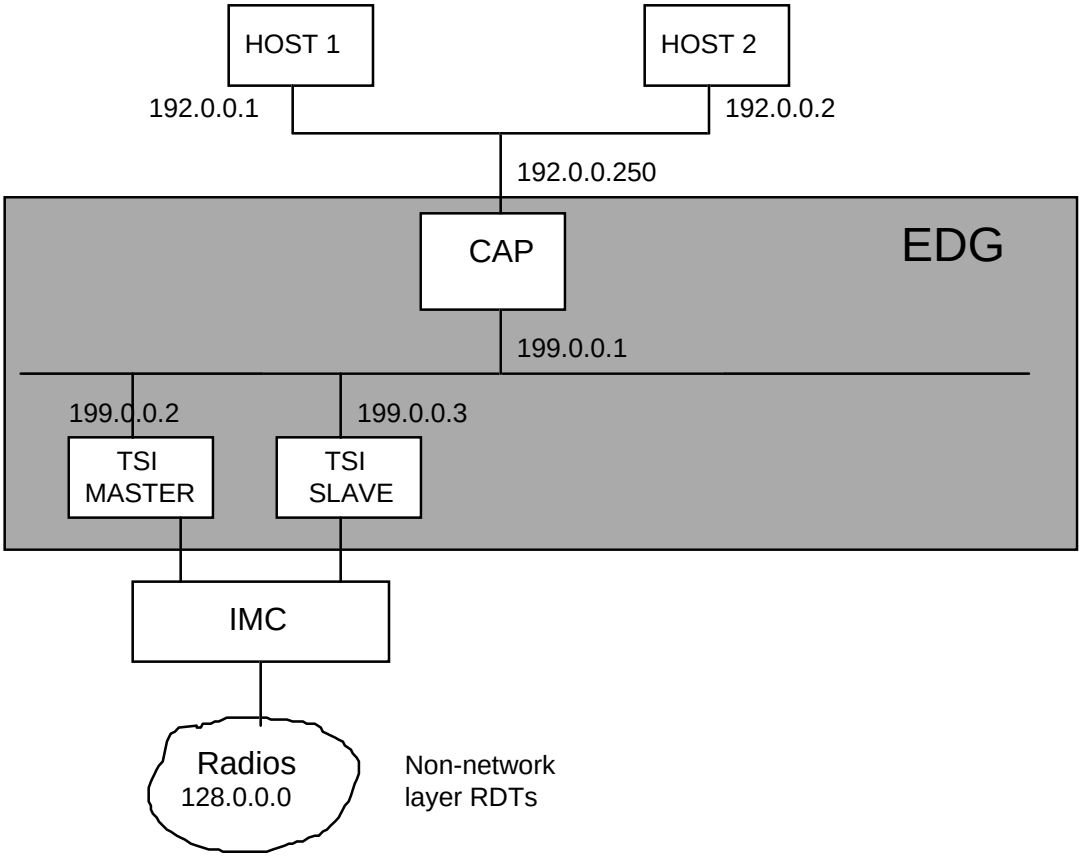
All other IDs are non-network layer type.

The table would then appear as follows:

[Network_Layer_RDT_Table]
u 300
u 1000-2000
g 0-999

SYSTEM CONFIGURATION EXAMPLES

CONFIGURATION EXAMPLE 1 - IP HOST WITH TSI MASTER/SLAVE

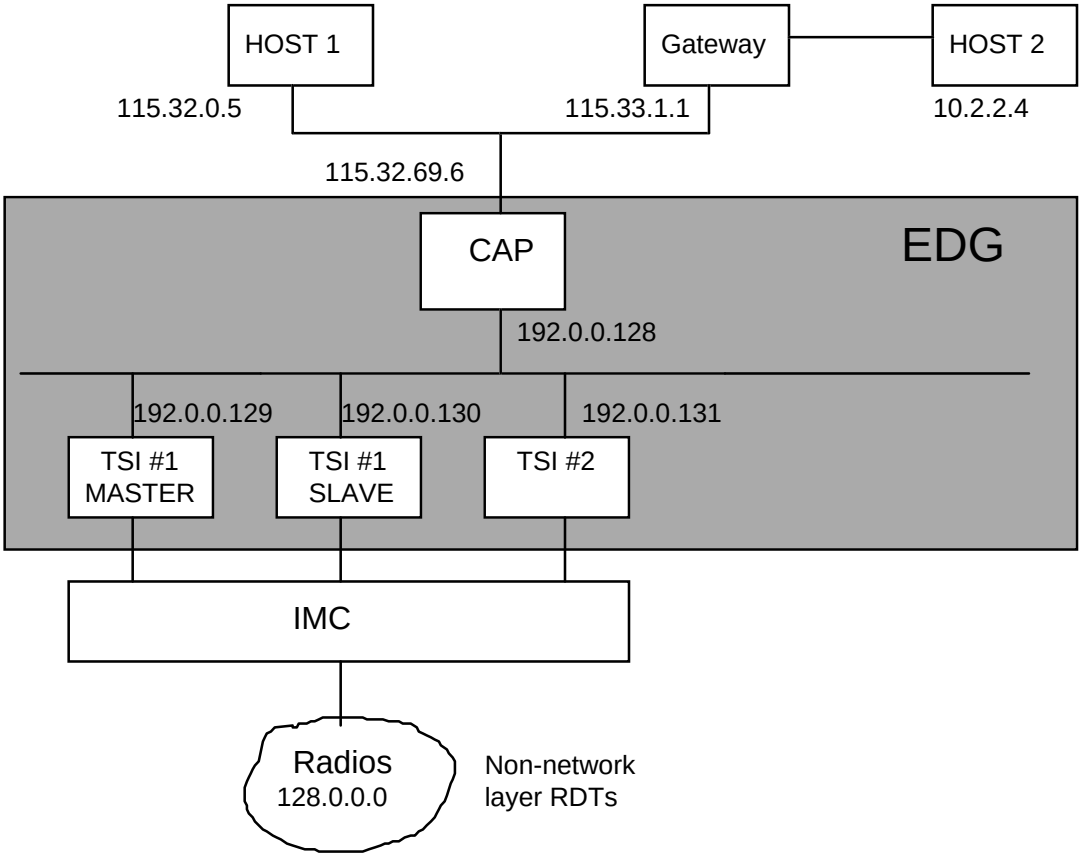


Example 1 represents a TSI Master/Slave configuration with two external IP hosts.
Since the IP hosts are on the CAP's local external network (they both have the same class C network ID as the CAP), a CAP external routing table is not required.

A possible SYSTEM.TXT file that would be used for this setup is as shown on the following page.

```
# Begin example 1 SYSTEM.TXT file

[Board 1]
Type    CAP
Load   01.02/loads/CAP.SX

[Board 2]
Type   TSI_Master
Load   01.02/loads/TSI.SX
TSI_EDACS_ID 16300
Port_Direction 1      In
Port_Direction 2      bi
Port_Direction 0      bi
Port_Direction 3      bi


[Board 3]
Type   TSI_Slave 2          # This slave's master is board 2.
Load   01.02/loads/TSI.SX
Port_Direction 0      bi
Port_Direction 1      bi
Port_Direction 2      bi
Port_Direction 3      bi

[IMC 1]
EDACS_IP_Network_ID    128.0.0.0

[IP]
CAP_Ext_Address  192.0.0.250
# System will use default internal board addresses 199.0.0.1 - 199.0.0.3

[IP_Map_ID_Table]
h       1              192.0.0.1
h       2              192.0.0.2
u       64-16000       128.0.0.64    # Single class B network for all units, groups
g       0-2047         128.0.64.0    # Assign groups in upper addresses of network

# End of example 1 SYSTEM.TXT file
```

CONFIGURATION EXAMPLE 2 - IP HOST WITH MULTIPLE TSI MASTERS



Example 2 is a slightly more complex configuration than example 1:

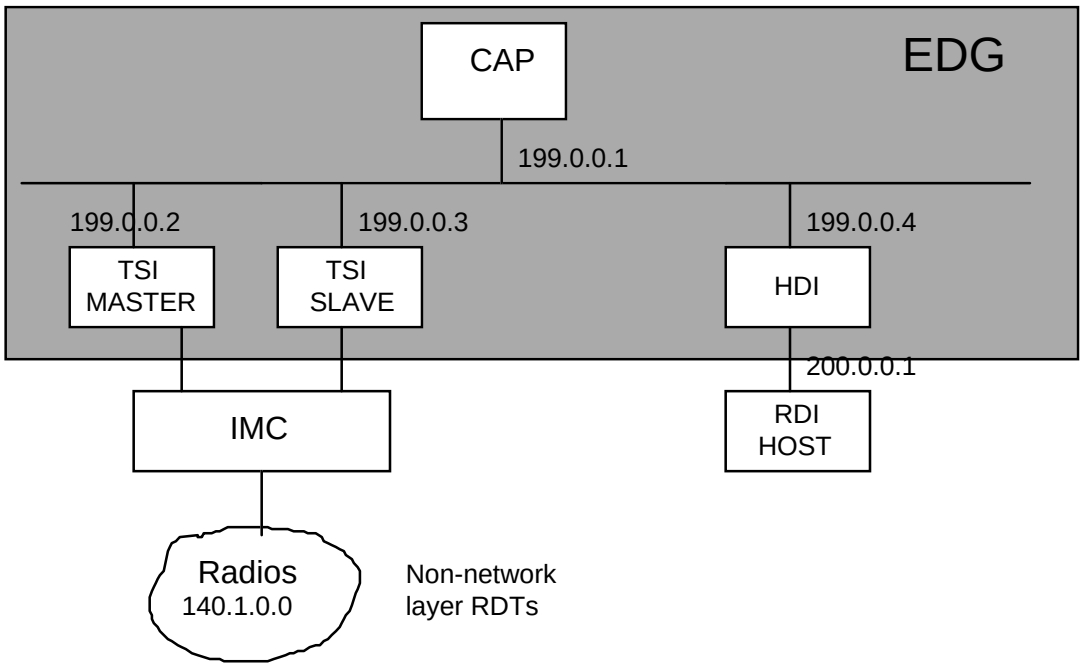
- There is now another TSI master in addition to the TSI Master/Slave boards. Since non-network layer RDTs are being used, it must be decided which host to assign to which TSI master. Since the TSI master/slave pair has eight communications channels (versus just four for the single TSI master), the IP host that will be generating the most data traffic would be assigned to the TSI master/slave pair.
- One of the IP hosts must be reached through a gateway. This will require the specification of a CAP external routing table.

A possible SYSTEM.TXT file that would be used for this setup is as shown on the following pages.

```
# Begin example 2 SYSTEM.TXT file

[Board 1]
Type    CAP
Load   01.02/loads/CAP.SX

[Board 2]
Type   TSI_Master
Load   01.02/loads/TSI.SX
port_direction 0      In
port_direction 1      bi
port_direction 2      bi
port_direction 3      bi
tsi_hosts             2       # Host 2 will be generating the most traffic
tsi_edacs_id          100

[Board 3]
Type   TSI_Slave 2          # This slave's master is board 2.
Load   01.02/loads/TSI.SX
Port_Direction 0      bi
Port_Direction 1      bi
Port_Direction 2      bi
Port_Direction 3      bi

[Board 4]
Type   TSI_Master
Load   01.02/loads/TSI.SX
TSI_Hosts             1       # Assign host 1 to this TSI
TSI_EDACS_ID      101
Port_Direction 0      BI
Port_Direction 1      BI
Port_Direction 2      Bi
Port_Direction 3      OUT

[IMC 1]
EDACS_IP_Network_ID    128.0.0.0
```

```
# Example 2 SYSTEM.TXT file continued

[IP]
int_network_id              192.0.0.0
int_board_address  1        192.0.0.128
int_board_address  2        192.0.0.129
int_board_address  3        192.0.0.130
int_board_address  4        192.0.0.131
CAP_Ext_Address   115.32.69.6

[IP_Map_ID_Table]
h       1               115.32.0.5
h       2               10.2.2.4
# Let the system assign default addresses to all the units and groups

[IP_CAP_ext_routing_table]
10.2.2.4                115.33.1.1

# End of example 2 SYSTEM.TXT file
```

CONFIGURATION EXAMPLE 3 - RDI HOST



Example 3 represents a TSI Master/Slave configuration with an RDI host. Since customers using an RDI host will not be concerned with explicitly assigning IP addresses, the default IP values for the internal EDG network, the IMC network, and the RDI hosts can be used. However, because there is no current default for the CAP external address, this value must still be specified. In this example, the address 1.0.0.1 was chosen at random. Its exact value is not significant since there is no connection to an IP network.

A possible SYSTEM.TXT file that would be used for this setup is as shown on the following page.

```
# Begin example 3 SYSTEM.TXT file

[Board 1]
Type    CAP
Load   01.02/loads/CAP.SX

[Board 2]
Type   TSI_Master
Load   01.02/loads/TSI.SX
TSI_EDACS_ID 16000
Port_Direction 1      In
Port_Direction 2      bi
Port_Direction 0      bi
Port_Direction 3      bi

[Board 3]
Type   TSI_Slave 2          # This slave's master is board 2.
Load   01.02/loads/TSI.SX
Port_Direction 0      bi
Port_Direction 1      bi
Port_Direction 2      bi
Port_Direction 3      bi

[Board 4]
Type   HDI
Load   01.02/loads/HDI.SX
HDI_Port_Hosts    0  1
HDI_Port_Hosts    1  1
Port_Direction 1      bi
Port_Direction 0      bi

[IP]
cap_ext_address     1.0.0.1         # This command must always be specified

# Note that the [IP_Map_ID_Table] and [IMC] sections need not be specified
# since system defaults will suffice.

# End of example 3 SYSTEM.TXT file
```

Ericsson GE Mobile Communications Inc.
Mountain View Road • Lynchburg, Virginia 24502

PRINTED IN U.S.A.