ARH6281

CONTROL DATA DISTRIBUTED COMMUNICATIONS NETWORK
(CDCNET)

RELEASE 1 EXTERNAL REFERENCE SPECIFICATION

LOG SUPPORT APPLICATION

PROPRIETARY NOTICE

	REVISION RECORD	
Revision	Description	Date
A	Orignal version, Release 1, WBS4	04/11/84
B	Updated to reflect design change	10/26/84
C	Final R1 update	10/21/85

Table of Contents

1.0 INTRODUCTION

1.0 INTRODUCTION

The Log Support Application (LSA) provides the features and services as described in the CDNA GDS for the Dependent Log M-E and the Dependent Alarm M-E. These features and services include:

- Generating Log and Alarm Data Units for all entities, processes, etc. in its CDCNET system.

- Forwarding certain Log Data Units to one or more Independent Log M-Es to be written to a log file.

- Forwarding certain Alarm Data Units to one or more operators and/or processes.

The LSA generates log and alarm data units for its local users and transmits the generated data units to one or more Independent Log M-Es to be written to a log file and/or to one or more Independent Alarm M-Es to be displayed at an operator's console.

For a CDCNET system to provide Dependent Log and Alarm Management the Dependent Log/Alarm M-E (known as the Log Support Application) must be present.

For a CDCNET system to provide Independent Log Management the Independent Log M-E must be configured as part of that system. An Independent Log M-E records log data on permanent storage and therefore the system that the M-E is configured to reside in must have access to permanent storage (disk). At least one CDCNET system in a catenet is required to have an Independent Log M-E.

For a CDCNET system to provide Independent Alarm Management, the Operator Support Application must be configured as part of that system. The Operator Support Application processes each alarm by displaying the alarm at the operator's console.

1.1 PURPOSE

This ERS addresses the design of the CDCNET Release 1.0 Log Support Application. The Log Support Application (LSA) provides logging and alarm services to all system entities, processes, etc., in the CDCNET system.

Every CDCNET system which is to log messages or send alarms must contain a Log Support Application. The Log Support Application supports the functionality required for the Dependent Log Management

---

1.0 INTRODUCTION
1.1 PURPOSE

---

and the Dependent Alarm Management. This functionality includes the
following:

- Receiving logging requests from local system
  entities, processes, etc.; in its CDCNET system.

- Generating Alarm and Log Data Units from the received
  logging requests.

- Transmitting Log Data Units to appropriate
  Independent Log M-E(s) for storing on a log file.

- Transmitting Alarm data units to appropriate
  Independent Alarm M-E(s) to be displayed at the
  operator console.

These services and features will be explained in detail in the
various sections of this document.

1.0 INTRODUCTION
1.2 REFERENCE DOCUMENTS

1.2 REFERENCE DOCUMENTS

     CDNA GDS   ARH4243

     CDNA Executive ERS   ARH4976

     Independent Log M-E ERS   ARH6903

     Generic Transport ERS   ARH6227

     Directory M-E ERS   ARH6384

     Network ERS, Section 10
       Network Operations   ARH6799

     Command M-E ERS   ARH5451

     Failure Management Around
     InterLayer Subroutine Calls

     Distributed Communications Network
     Software Program Interface Handbook

2.0 FEATURE/SERVICE OVERVIEW

2.0 <u>FEATURE/SERVICE OVERVIEW</u>

The following sections present an overview of the Log Support Application software.

2.1 <u>FEATURE/SERVICE OF THE LSA</u>

2.1.1 LOGGING

The Log Support Application provides a Log Data Unit Generation/Transmission Service to its users. Log Data Units are defined as data units which carry log messages between a Log Support Application and an Independent Log M-E. The LSA service is accessed via a direct call to an LSA provided program interface. The LSA adds appropriate system information to the user provided log message (i.e., time stamp, source system address, etc.), and sends the message to the appropriate system(s) for storing on a log file.

2.1.2 ALARMS

The Log Support Application provides an Alarm Data Unit Generation/Transmission service to its users. Alarm Data Units are defined as data units which carry alarm messages between the Log Support Application and an Independent Alarm M-E. Alarming services are accessed via the same direct call interface as logging services. The LSA adds appropriate system information to the user provided message (i.e., time stamp, source system address, etc.), and sends the message to the appropriate system(s) to be displayed at the operator's console.

2.1.3 QUEUING OF USER REQUESTS

Before the configuration of the LSA is complete (the state of the DI is being configured) each message to be logged is placed in the LSA initialization queue. Once the configuration of the LSA is complete (the state of the DI is operational) each message in the LSA initialization queue which is enabled as a log message and/or as an alarm is transmitted to the destination Independent Log M-E and/or Independent Alarm M-E respectively. Each message to be logged after the DI is in an operational state is sent to the LSA via intertask message.

The LSA places each message it receives in the Log Preserve Queue. The Log Preserve Queue is a circular queue which contains the last 15 log messages received by the LSA. This queue is provided by the LSA for debugging purposes.

2.0 FEATURE/SERVICE OVERVIEW
2.1.3 QUEUING OF USER REQUESTS

The LSA requests a title translation for each log group it supports. A logical connection table is kept for each indication received from the Directory M-E. Prior to receiving a dir_indication_done, indicating that the title translation is complete, each log message received by the LSA which is enabled is queued in the log group queue. Once all Directory M-E indications and the dir_indication_done have been received, the log messages queued in the log group queue are place in each logical connection queue of the highest priority. An attempt is then made to establish a Generic Transport connection to each destination Independent Log M-E for each logical connection of the highest priority. If a Generic Transport connection already exists to the destination Independent Log M-E the existing connection is used. Once the Generic Transport connection is established all log messages queued in the logical connection queue are transmitted to the destination Independent Log M-E.

The LSA requests a title translation for each alarm group it supports. An Alarm Connection Table exists for each indication received from the Directory M-E. Each log message received by the LSA, which is enabled as an alarm, is queued in each Alarm Connection Table Queue. A Generic Transport connection is established whenever an alarm message is to be delivered to a destination Independent Alarm M-E. When the connection is available all alarm messages queued are immediately transmitted to the destination Independent Alarm M-E. Each alarm received by the LSA, while the connection is established, is transmitted immediately to the destination Independent Alarm M-E. If the Generic Transport Connection remains inactive for 10 seconds the connection is destroyed.

Each connection queue supported by the LSA has a maximum queue size of 250 messages. If any LSA queue size is exceeded, one additional log message is generated by the LSA itself and queued, indicating that user log or alarm messages were discarded for that particular queue. Any future messages destined for that queue are discarded until the overflow condition is resolved. The catenet will continue to function, but transmission of Log Data Units for the affected connection will be suppressed (i.e., the LSA Log Data Unit Generation Service will continue to accept log information, but the information will simply be discarded). This approach localizes the knowledge of the absence of the Independent Log M-E and/or Independent Alarm M-E.

2.0 FEATURE/SERVICE OVERVIEW
2.1.4 LOG/ALARM MESSAGE MAPS

2.1.4 LOG/ALARM MESSAGE MAPS

Two types of message maps are provided for the logging and alarming
environments, the permanent message maps and the temporary message
maps. The permanent message maps initially contain those messages
which were defined as a result of logging and alarming configuration
commands being executed from the configuration file during system
configuration (via the DEFINE_SOURCE_LOG_GROUP command for log
messages and the DEFINE_SOURCE_ALARM_MESSAGE command for alarms).
The temporary message maps always contain the current set of
messages which are to be logged or sent as alarms. Just after the
configuration of the DI is complete, the temporary message map is
identical to the permanent message map. The LSA maintains both a
permanent and a temporary list of the log messages selected for each
log group and both a permanent and a temporary list of the alarm
messages selected for each system.

The CHANGE_SOURCE_LOG_GROUP command is provided to allow the
operator to select additional messages to be logged or to disable
some of the messages which are currently selected (e.g., to collect
additional statistics or to disable reporting of certain
statistics). For alarm messages, the DEFINE_SOURCE_ALARM_MESSAGE
command can be issued after the system configuration is complete to
add alarm messages or the CANCEL_SOURCE_ALARM_MESSAGE command can be
used to cancel selection of messages previously sent as alarms. The
temporary message lists are updated as a result of these commands
and the changes will remain in effect until the operator chooses to
return to using the permanent message lists or the system is
reloaded. If the operator would like to retain the changes made to
the temporary message maps, a retain command can be used (the
RETAIN_SOURCE_LOG_GROUP command for log message maps and the
RETAIN_SOURCE_ALARMS command for alarm message maps). The temporary
changes made to the set of messages to be logged or sent as alarms
will become permanent (retained until the next time the system is
reloaded). Changes can be made permanent across DI resets by making
the necessary changes to the configuration file. Further changes
can then be made to the temporary message set, if desired.

At any time, the operator may return to logging the permanent set of
messages by using a restore command (the RESTORE_SOURCE_LOG_GROUP
command for log message maps and the RESTORE_SOURCE_ALARMS command
for alarm message maps). These commands restore the set of messages
to be logged or sent as alarms to those in the permanent message
list (those selected during configuration via the
DEFINE_SOURCE_LOG_GROUP and DEFINE_SOURCE_ALARM_MESSAGE commands or
those selected when the last retain command was issued).

For Release 1, the retain and restore commands will not be
supported.

2.0 FEATURE/SERVICE OVERVIEW
2.1.5 NETWORK OPERATOR / CONFIGURATION INTERFACE

2.1.5 NETWORK OPERATOR / CONFIGURATION INTERFACE

In correspondence to the services mentioned above, the LSA provides
a network operator/configuration interface feature (i.e., Command
Processor interface to the LSA). The configuration of the LSA
logging and alarming environments are completed solely with the use
of Command Processors. Command Processors are provided to allow the
operator to control the logging and alarming environments of each
system on an individual basis.

The initial configuration of the LSA logging environment and the LSA
alarm environment is constructed from commands executed via the
configuration file. Once the Configuration Procurer has completed
processing the commands in the configuration file, it sets the
system state from "being configured" to "operational" and then logs
a message indicating that the configuration of the DI is complete.
Once the system state is set to "operational" and the LSA has been
started by System Ancestor as part of the minimum set, logging and
alarming services are configured. This means that connections will
be established to the appropriate destination Independent Log M-Es
and Independent Alarm M-Es for the purpose of transmitting Log Data
Units and Alarm Data Units to the respective destination systems.

2.1.6 LOGGING VECTOR

A composite logging vector is maintained by the LSA which identifies
all log and alarm messages enabled within the DI. The LSA provides
a function (log_message_enabled) which allows software components to
determine if a particular log message is enabled or not. If the log
message is enabled the software component should generate the log
message and send it to the LSA via the log_request common
subroutine. If the message is not enabled the software component
should not build the log message. This function eliminates the the
overhead of building and sending log messages to the LSA which are
not enabled.

2.0 FEATURE/SERVICE OVERVIEW
2.2 FUNCTIONAL RELATIONSHIP

2.2 FUNCTIONAL RELATIONSHIP

The LSA provides two user interfaces which are accessible to all
system entities, processes, etc., in the local CDCNET system (refer
to diagram 2.2-1):
  . log_message_enabled
  . log_request

These interfaces are provided via a direct call mechanism. The
log_message_enabled procedure allows the caller to determine if a
specific log message should be generated. For a message which is
generated, the Log and Alarm Data Unit Generation/Transmission
Services provided by the LSA are utilized by calling the log_request
procedure. LSA determines the destination of the message provided
by the caller and sents it to the appropriate destinations.

```
+-------------+------------------------------------------+
|             |                                          |
|             |         -----           LOG     ,        |
|          user i/f              SUPPORT          -----  |
|          ============>         APPLICATION      <====== Command
|         (log_request)                           ------  Processors
|      (log_message_enabled) _                            |
|             -----                                       |
| OTHER                                                   |
| SYSTEM      |                   |                        |
| ENTITIES,   | Log Data Unit     | Alarm Data Unit        |
| PROCESSES,  | Generation /      | Generation /           |
| ETC.        | Transmission      | Transmission           |
|             | Services          | Services               |
|             |     :             |     :                  |
|         +---| :  |----+----|  :  |------+               |
|         |   V          |        V       |               |
|         |         Transport             |               |
|         |     ↑              ↑           |               |
+---------+---| :  |----------| :  |------+               |
|         |   :          |     :           |               |
|         |   :          |     :           |               |
|         |   :          |     :           |               |
|     +---| :  |---------| :  |----+                       |
|     |   V          |        V    |                       |
|     |        Transport           |                       |
|     |   ↑            ↑            |                       |
|  +--| :  |---+-----|  :  |------+                         |
|  |  V        |         V        |                         |
|  | Independent | Independent    |                         |
|  | Log M-E     | Alarm M-E      |                         |
+--+-------------+----------------+                         +
```
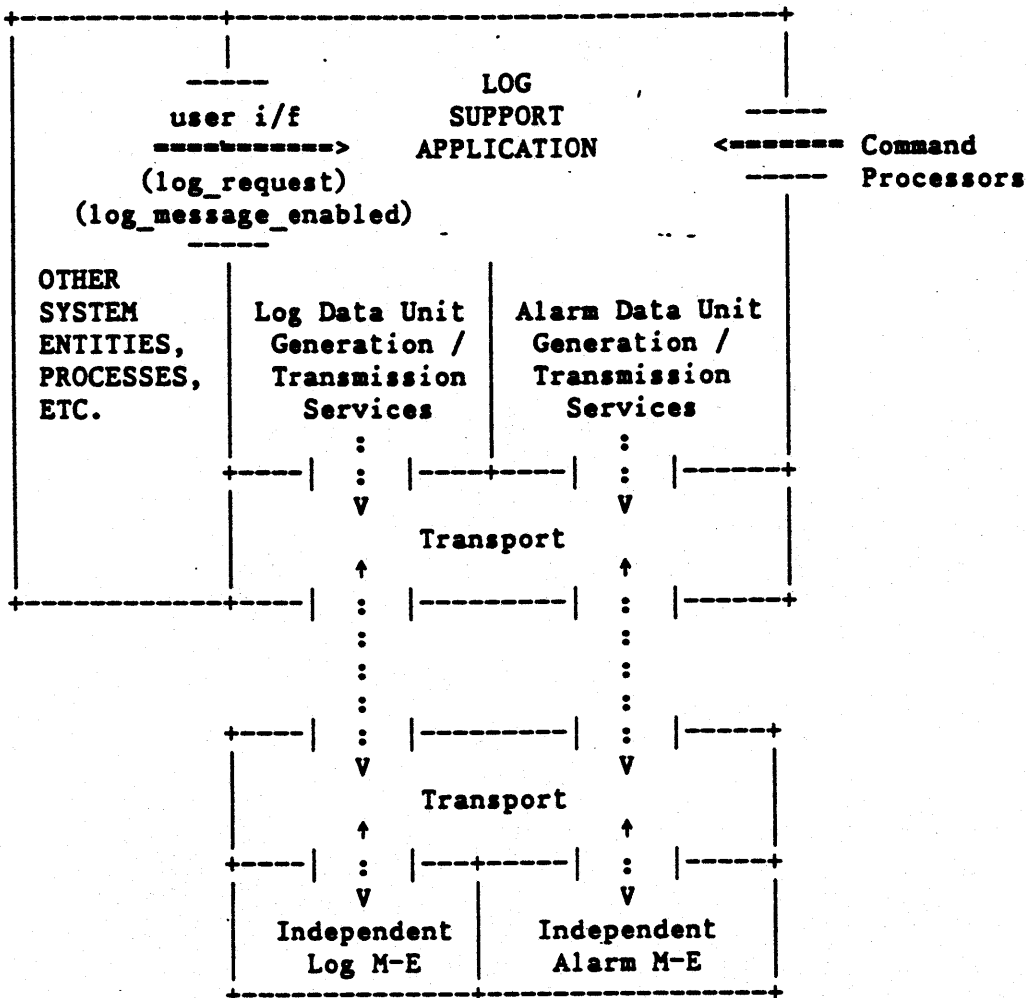
Figure 2.2-1 Overview Diagram

2.0 FEATURE/SERVICE OVERVIEW
2.2 FUNCTIONAL RELATIONSHIP

The LSA provides a Logging and Alarm Data Unit Transmission service
via the use of Transport connections.

Logging and alarm configuration commands identify the log groups and
alarm groups to be supported by the LSA. Log groups and alarm
groups identify the destination Independent Log M-E(s) and
destination Independent Alarm M-E(s) each log message received by
the LSA is to be sent to. Likewise, logging configuration commands
identify which messages are to be enabled (sent to the destination
M-Es) for each log group or alarm group supported by the LSA.

Each Independent Log M-E and Independent Alarm M-E is qualified as
belonging to a specific group by uniquely defining a group title.
The title for each log group is defined as "$I_LOG_ME_'n'" where 'n'
uniquely identifies a specific log group. Likewise, the title for
each alarm group is defined as "$I_ALARM_ME_'m'" where 'm' uniquely
identifies a specific alarm group.

Once the Configuration Procurer has processed all commands in the
System Configuration File, the logging and alarming environment is
considered to have been defined. Then, the LSA utilizes local
Directory M-E services to request a passive title Translation for
each "I_LOG_ME_'n'" title which has been placed in its Correspondent
Log Data Store by the LSA support Command Processors. The title
translations received by the Log Support Application for a given
"$I_LOG_ME_'n'" title will always return the address of the
Independent Log M-E and its priority. When multiple Independent Log
M-E systems exist that support the same log group, Transport
connections are established to each Independent Log M-E with the
highest priority received on the title translation. If the
connection already exists from the Log Support Application to the
particular Independent Log M-E, the existing connection is utilized.
This allows multiple log groups associated with the same Log Support
Application/Independent Log M-E pair to utilize the same Transport
Connection. Transport connections between the Log Support
Application and Independent Log M-E remain active until a change in
the log group definition has caused a particular Log Support
Application and Independent Log M-E to no longer be associated with
a common log group.

Likewise, the LSA utilizes local Directory M-E services to request a
passive title Translation for each "I_ALARM_ME_'m'" title which has
been placed in its Correspondent Alarm Data Store by the LSA support
Command Processors. For each translation received a transient
Transport connection is established to that Independent Alarm M-E.
If the connection already exists from the Log Support Application to
the particular Independent Alarm M-E, the existing connection is
utilized. This allows multiple alarm groups associated with the
same Log Support Application/Independent Alarm M-E pair to utilize

----
2.0 FEATURE/SERVICE OVERVIEW
2.2 FUNCTIONAL RELATIONSHIP
----

the same Transport Connection. Transport connections between the Log Support Application and Independent Alarm M-E are disconnected after 10 seconds of inactivity.

2.3 UTILIZED EXTERNAL INTERFACES

This section identifies and describes the utilized external interfaces of the LSA. Besides the interfaces described, several of the Log Support Application modules use EXEC services for the buffer management and intertask messages. In view of the standard nature of these interfaces, they are not described below.

2.3.1 TRANSPORT/LSA INTERFACE

The LSA establishes Transport connections for transmitting Log Data Units to the appropriate Independent Log M-E(s) and Alarm Data Units to the appropriate Independent Alarm M-E(s). Generic Transport services are provided by a direct call interface.

2.3.2 DIRECTORY M-E/LSA INTERFACE

The Directory M-E provides the title registration and translation services required by Independent Log M-E. Directory services are provided by a direct call interface.

The Independent Log M-E requests a title translation for "Generic Transport" for the purpose of opening a Generic Transport SAP.

The Directory M-E services are also used by the LSA to obtain Translations for the Independent Log M-E(s) and the Independent Alarm M-E(s). After the configuration of the LSA logging and alarming environments are complete, the LSA requests a passive title translation for each configured log group and alarm group supported by the LSA.

2.3.3 STATISTICS MANAGER/LSA INTERFACE

The Statistics Manager provides assistence in the collection of statistics by calling the user supplied procedures to generate log messages containing the statistics. It keeps track of which statistics are to be collected and reported and at what interval, and maintains statistics data structures. The LSA opens a statistics SAP, collects the loging statistics and provides a procedure which the Statistics Manager will call to report the

2.0 FEATURE/SERVICE OVERVIEW
2.3.3 STATISTICS MANAGER/LSA INTERFACE

statistics.

3.0 FEATURE/SERVICE DESCRIPTION

3.0 FEATURE/SERVICE DESCRIPTION·

3.1 LOG DATA UNIT GENERATION/TRANSMISSION SERVICE

The LSA receives logging requests from its users via a direct call to the LSA provided program interface. Upon receiving a message, the LSA determines if the message provided is enabled as either a log message or as an alarm and if not enabled, it is discarded. If the message is enabled for logging, the LSA utilizes the Generate Log Data Unit service which adds appropriate system information (i.e., time stamp, source system address, etc.) to the user provided message to form a Log Data Unit. The Log Data Units are used to carry messages from the LSA to the destination Independent Log M-E(s).

After a Log Data Unit is generated for a specific log message, the Log Data Unit is transmitted from the LSA to each Independent Log M-E that supports the log group for which the log message is enabled, via the Log Data Unit Transmission service. The Log Data Unit Transmission service utilizes Transport connection~ to transmit the data units.

3.2 ALARM DATA UNIT GENERATION/TRANSMISSION SERVICE

The LSA receives logging requests from its users via a direct call to the LSA provided program interface. Upon receiving a message, the LSA determines if the message provided is enabled as either a log message or as an alarm and if not enabled, it is discarded. If the message is enabled as an alarm, the LSA utilizes the Generate Alarm Data Unit service which adds appropriate system information (i.e., time stamp, source system address, etc.) to the user provided message to form an Alarm Data Unit. The Alarm Data Units are used to carry messages from the LSA to the destination Independent Alarm M-E(s).

After an Alarm Data Unit is generated for a specific message, the Alarm Data Unit is transmitted from the LSA to each Independent Alarm M-E via the Alarm Data Unit Transmission service. The Alarm Data Unit Transmission service utilizes realiable Transport connections to transmit the data units.

3.0 FEATURE/SERVICE DESCRIPTION
3.3 EXTERNAL INTERFACES

3.3 EXTERNAL INTERFACES

3.3.1 LOG MESSAGE ENABLED INTERFACE

The first step in generating a log message is to determine if the
message has been selected for logging or as an alarm for the local
system. The "log_message_enabled" function provides this
information to all system entities, processes, etc., in the local
CDCNET system.

Users must include in their calling module a *CALLC LSXLOGR which
defines the the log_message_enabled function.

Entry Point Name
   FUNCTION [XREF] log_message_enabled ({
        message_number: log_msg_id_type;
        priority: log_priority
        ): boolean;

Description of Input Parameters

        Name: Description:

message_number   The unique log ID of the log message for which the
                  inquiry is being made.

     priority   The priority of the log message. Log_priority is
                  defined as an ordinal with the values log_critical,
                  log_high, log_medium and log_low.

Returns

     log_message_enabled   A boolean value indicating whether or not
                            the message is enabled. The value is
                            normally set to TRUE if the message is
                            enabled and to FALSE if the message is
                            disabled. However, if memory or buffer
                            space in the DI is low, the flag may be
                            set to FALSE even if the message is
                            enabled (the decision is based on the
                            priority of the log message and the
                            severity of the memory and buffer space
                            shortage).

3.0 FEATURE/SERVICE DESCRIPTION
3.3.1 LOG MESSAGE ENABLED INTERFACE

If the "log_message_enabled" function returns a value of FALSE, the
message need not be generated, thus eliminating the overhead of
generating the Management Data Units (MDUs) containing the template
identifiers and variable parts for log messages which LSA would
discard when received via log_request.

3.3.2 LOG REQUEST PROGRAM INTERFACE.

Once the Management Data Units (MDUs) containing the template
identifiers and variable parts of a log message have been generated,
the message needs to be written to the appropriate log files and/or
delivered to the network operators as an alarm. The "log_request"
interface provides this service to all system entities, processes,
etc., in the local CDCNET system.

Users must include in their calling module a *CALLC LSXLOGR which
defines the XREF declaration for the LSA interface routine. It also
defines the parameter attributes of the procedure.

Entry Point Name
  PROCEDURE [XREF] log_request ({
        message_id: log_msg_id_type;
    VAR message: buf_ptr);

Description of Input Parameters

    Name: Description:

message_id   The unique log ID of the message to be logged.

   message   The address of the buffer containing the template
             identifiers and the variable parts of the message in
             MDU format.

             If the message buffer contains any variable parts which
             are not preceded by a template identifier, a template
             id equal to zero is used. A zero template id is a
             special id which is used to indicate that a template
             has not been defined for the associated variable data.
             A default template is used.

             If the message has no variable parts, there must be one
             or more template identifiers which define the fixed
             text.

3.0 FEATURE/SERVICE DESCRIPTION
3.3.2 LOG REQUEST PROGRAM INTERFACE

Returns

The log_request procedure will discard the log message provided by
the caller (user) for any of the following reasons.

1. The DI buffer state is congested.

2. The DI memory state is congested.

3. The log message id specified is out of range.

4. The log message is not enabled as a log message or as
    an alarm message in the LSA logging or alarming environment.

The log_request procedure returns a NIL value to the caller via the
log_message pointer provided. The copy of the log_message provided
now belongs to the Log Support Application.

3.4 NETWORK OPERATOR INTERFACE

The Network Operator Interface shown on figure 2.2-1 as the Command
Processor interface, provides the network operator   with   the
capability to control the LSA logging and alarm environments.
Operator commands allow the network operator the capability to
specify system(s) which are to receive Log Data Units and/or Alarm
Data Units from the LSA, as well as the capability to filter log
messages and alarm messages. The operator can filter log messages
or alarm messages on an individual system basis by turning on or off
the transmission of a particular message.

The network operator interface to the LSA is provided through the
use of the local system Command M-E. The Command M-E is required to
start the appropriate Command Processor to execute the operator
command requested, and to route a generated response back from the
LSA support Command Processors.

3.4.1 COMMAND INTERFACE

3.4.1.1 Logging Command Interface

3.4.1.1.1 DEFINE_SOURCE_LOG_GROUP | DEFSLG

This command defines one or more log groups that the Dependent Log
M-E belongs to and the messages to be logged for those groups.

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.1.1 DEFINE_SOURCE_LOG_GROUP | DEFSLG
----

define_source_log_group
    log_group, lg: list 1 .. 15 of name = catenet
    message_number, mn: list 1 .. 31 range of integer (1 .. 32999)

    log_group | lg: This parameter specifies one or more
                           log groups to which the Dependent
                           Log M-E will belong. The default and
                           only valid log group name for R1 is
                           CATENET.

    message_number | mn: This parameter specifies one or more
                           log message numbers to be defined for
                           the log groups specified.

RESPONSES:

--INFORMATIVE-- Source log groups defined.
        Indicates that the log groups specified were defined
        and that the selected messages are enabled for logging.

--WARNING-- Source log groups defined, no message numbers specified.
        Indicates that a log groups has been defined with no
        message numbers enabled.

--ERROR-- Duplicate log group <name> specified.
        Indicates that the same log group name was specified
        more than once in the list of log groups to be defined.
        No log groups are defined.

--ERROR-- Source log group <name> is not a known community name.
        The log group name specified is not in the list of
        communities supported in the DI (For R1 the only valid
        community is CATENET).

--ERROR-- Source log group <name> already defined.
        The log group name specified has already been defined
        with a different set of log messages. If a change in
        message numbers to be logged is desired, the
        change_source_log_group command must be used.

--FATAL-- Not enough memory is currently available for required table space.
        This message is returned if the LSA cannot obtain
        the memory required for the associated log group data
        store.

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.1.2 CHANGE_SOURCE_LOG_GROUP | CHASLG

3.4.1.1.2 CHANGE_SOURCE_LOG_GROUP | CHASLG

This command changes the log messages defined for one or more of the log groups that a Dependent Log M-E belongs to. All message numbers specified by the add_message_number parameter will be defined and then all messages numbers specified by the delete_message_number parameter will be cancelled. Changes made by this command remain in effect until the next DI reload.

```
change_source_log_group
     log_group, lg: list 1 .. 15 of name = catenet
     add_message_number, amn: list 1 .. 31 range of integer (1 .. 32999)
     delete_message_number, dmn: list 1 .. 31 range of integer (1 .. 32999)
```

     log_group | lg: This parameter specifies one or more log groups for which the log messages enabled are being changed. The default and only log group name valid for R1 is CATENET.

     add_message_number | amn: This parameter specifies one or more log message numbers to be defined for the log groups specified.

     delete_message_number | dmn: This parameter specifies one or more log message numbers to be cancelled for the log groups specified.

RESPONSES:

--INFORMATIVE-- Source log groups changed.
          Indicates that the log messages defined for the specified log groups have changed. Message numbers were defined or cancelled as specified by the add_message_number and delete_message_number parameters.

--WARNING-- No message numbers specified.
          Indicates that no change was made to the log messages defined for the specified log groups, because no message numbers were specified by either the add_message_number or delete_message_number parameter.

--ERROR-- Duplicate log group <name> specified.
          Indicates that the same log group name was specified more than once in the list of log groups to be changed. No log groups are changed.

--ERROR-- Source log group <name> is not defined.
          Indicates that no change was made to the log messages defined, because the specified log group is not defined.

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.1.2 CHANGE_SOURCE_LOG_GROUP | CHASLG

3.4.1.1.3 CANCEL_SOURCE_LOG_GROUP | CANSLG

This command deletes one or more log groups from the list of groups
that a Dependent Log M-E belongs to.

cancel_source_log_group
    log_group, lg: list 1 .. 15 of name = catenet

    log_group | lg: This parameter specifies one or more log
          groups to be deleted. The default and only log group
          name valid for R1 is CATENET.

RESPONSES:

--INFORMATIVE-- Source log groups cancelled.
        Indicates that the log groups specified were deleted.

--WARNING-- No source log groups defined.
        Indicates that there are no log groups to cancel.

--WARNING-- Specified source log groups cancelled.
Source log group <name> was not defined.
        Indicates that the log groups were cancelled, but one or
        more of the specified log groups was not defined (prior
        to the cancel).

--ERROR-- Duplicate log group <name> specified.
        Indicates that the same log group name was specified
        more than once in the list of log groups to be cancelled.
        No log groups are cancelled.

3.4.1.1.4 DISPLAY_SOURCE_LOG_GROUP | DISSLG

This command displays the log groups that the Dependent Log M-E
belongs to and the messages to be logged for each group.

display_source_log_group

RESPONSES:

--INFORMATIVE--
                Source Log Groups
Log Group <name>
Log message numbers defined:
    <list of log messages defined>

CONTROL DATA PRIVATE

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.1.4 DISPLAY_SOURCE_LOG_GROUP | DISSLG

All of the log groups supported by the Dependent
Log M-E and the messages defined for each are
listed. The text following the "Source Log
Groups" header is repeated for each log group
supported by the LSA. If no message numbers
are defined for a particular log group, the text

No log message numbers defined.

replaces the "Log message numbers defined" portion
of the response.

--INFORMATIVE-- No source log groups defined.
        There are no log groups currently defined for the
        Dependent Log M-E.

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.2 Alarm Command Interface

3.4.1.2 <u>Alarm Command Interface</u>

3.4.1.2.1 DEFINE_SOURCE_ALARM_GROUP | DEFSAG

This command defines one or more alarm groups that the Dependent
Alarm M-E (LSA) belongs to. For Release 1, if any alarm messages
are defined by the DEFINE_SOURCE_ALARM_MESSAGE command, the
Dependent Alarm M-E will by default belong to the alarm group
CATENET.

define_source_alarm_group
    alarm_group, ag: list 1 .. 15 of name = catenet

    alarm_group | ag: This parameter specifies an alarm group
                        or list of alarm groups that the
                        Dependent Alarm M-E will belong to. The
                        default and only alarm group name valid
                        for R1 is CATENET.

RESPONSES:

--INFORMATIVE-- Source alarm groups defined.
        Indicates that the alarm groups specified were defined.

--ERROR-- Duplicate alarm group <name> specified.
        Indicates that the same alarm group name was specified
        more than once in the list of alarm groups to be defined.
        No alarm groups are defined.

--ERROR-- Source alarm group <name> is not a known community name.
        The alarm group name specified is not in the list of
        communities supported in the DI. For R1, the only valid
        community is CATENET.

--ERROR-- Source alarm group <name> already defined.
        The alarm group name specified has already been defined.

--FATAL-- Not enough memory is currently available for required table space.
        This message is returned if the LSA cannot obtain
        the memory required for the associated alarm group data
        store.

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.2.2 CANCEL_SOURCE_ALARM_GROUP | CANSAG

3.4.1.2.2 CANCEL_SOURCE_ALARM_GROUP | CANSAG

This command cancels an alarm group that the Dependent Log M-E belongs to.

cancel_source_alarm_group
    alarm_group, ag: list 1 .. 15 of name = catenet

    alarm_group | ag: This parameter specifies the alarm group
                        or list of alarm groups to be cancelled
                        from the Dependent Alarm M-E alarm
                        environment. The default and only alarm
                        group name valid in R1 is CATENET.

RESPONSES:

--INFORMATIVE-- Source alarm groups cancelled.
        Indicates that the alarm groups specified were cancelled.

--WARNING-- No source alarm groups defined.
        Indicates that there are no alarm groups to cancel.

--WARNING-- Specified source alarm groups cancelled.
Source alarm group <name> was not defined. -
        Indicates that all alarm groups have been deleted from
        the LSA alarming environment, but one or more of the
        alarm groups were not defined (prior to the cancel).

--ERROR-- Duplicate alarm group <name> specified.
        Indicates that the same alarm group name was specified
        more than once in the list of alarm groups to be cancelled.
        No alarm groups are cancelled.

3.4.1.2.3 DEFINE_SOURCE_ALARM_MESSAGE | DEFSAM

This command defines one or more alarm messages within the alarmimg
environment. If more than one DEFSAM command is issued, the alarm
messages defined will be those specified on the command, in addition
to those specified on any previous DEFSAM commands.

define_source_alarm_message
    message_number, mn: list 1 .. 31 range of integer 1 .. 32999
        = $REQUIRED

    message_number | mn: This parameter specifies one or
                           more alarm message numbers to be
                           defined. This parameter is required.

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.2.3 DEFINE_SOURCE_ALARM_MESSAGE | DEFSAM

RESPONSES:

--INFORMATIVE-- Source alarm messages defined.
        Indicates that the alarm messages have been defined.

--FATAL-- Not enough memory is currently available for required table space.
        This message is returned if the LSA cannot obtain
        the memory required for the associated alarming message
        table.

3.4.1.2.4 CANCEL_SOURCE_ALARM_MESSAGE | CANSAM

This command cancels one or more alarm messages within the alarming
environment.

cancel_source_alarm_message
    message_number, mn: list 1 .. 31 range of integer 1 .. 32999
      - $REQUIRED
    message_number | mn: This parameter specifies one or
                          more alarm message numbers to be
                          cancelled. This parameter is required.

RESPONSES:

--INFORMATIVE-- Source alarm messages cancelled.
        Indicates that the alarm messages specified have been
        cancelled.

3.4.1.2.5 DISPLAY_SOURCE_ALARMS | DISSA

This command displays the alarm groups supported by the Dependent
Alarm M-E (LSA) and to display the set of alarm messages which are
defined for the system.

display_source_alarms

RESPONSES:

--INFORMATIVE--
                Source Alarms
Alarm groups:

3.0 FEATURE/SERVICE DESCRIPTION
3.4.1.2.5 DISPLAY_SOURCE_ALARMS | DISSA

        <list of alarm groups supported by the LSA>
Alarm message numbers defined:
        <list of defined message numbers>

        If no source alarm messages are defined, the source
        alarm message portion of the message is replaced by:

    No alarm message numbers defined.

--INFORMATIVE-- No source alarm groups defined.
        Indicates that no source alarm groups are defined.

4.0 PERFORMANCE

4.0 PERFORMANCE

4.1 NUMBER OF INSTRUCTIONS/MEMORY

The Log Support Application is a Cybil module which consists of both the Dependent Log M-E and the Dependent Alarm M-E. Each of these components has a set of supporting command processors and key data structures. In addition, the Log Support Application provides an INLINE function (log_message_enabled) to determine if a log message is enabled in the DI. Likewise, the LSA provides an interface which is directly called to log a message (log_request). Each of these interfaces is utilized by other entities, processes, etc. in the DI. A summary of the Log Support Application memory requirements and the number of instructions used by the LSA common subroutines follows.

Routine	Memory requirements
Dependent Log M-E	2 K
* supporting command processors	1.5 K
* each log group supported	82 + (n * 58) bytes
* each Transport Connection established	22 bytes
Dependent Alarm M-E	2 K
* supporting command processors	1.5 K
* Alarm Data Store	35 bytes
* each alarm group supported	52 bytes
* each Transport Connection established	52 + (m * 26) bytes

Routine	Number of Instructions
LSA Common Interfaces	
log_message_enabled	20
log_request	100

(Where n = the number of unique Directory M-E indications received and
       m = the numbewr of IAME's in the Catenet.)

4.0 PERFORMANCE
4.2 LSA STATISTICS

4.2 <u>LSA STATISTICS</u>

The Log Support Application records the following statistics:

1. The number of log/alarm messages received.

2. The number of log messages transmitted.

3. The number of alarm messages transmitted.

4. The number of disabled log/alarm messages discarded.

5. The number of log messages discarded due to queue limit.

6. The number of alarm messages discarded due to queue limit.

7. The number of log/alarm messages discarded due to
   insufficient buffers or memory.

5.0 FINITE STATE MACHINE

5.0 <u>FINITE STATE MACHINE</u>

The interface between the Independent Log M-E and the Dependent Log
M-E as well as the interface between the Independent Alarm M-E and
the Dependent Alarm M-E is driven through operator commands and
titles registered with the Directory M-E. Therefore, no FSM is
required.

6.0 LOG MESSAGES

6.0 LOG MESSAGES

The purpose of this log message is to indicate that a log group specified on the define_source_log_group command was successfully defined and that the selected messages are defined for logging.

Log Message ID

```
CONST
    ls1$log_group_defined = min_log_message_id + 409;
```

MASK		LOG _ MESSAGE _ BUFFER		
fixed text		type	value	description
see mask1 below		char octet	1..31 chars	Name of log group defined
see mask2 below		bin_int	list of integer	List of defined message numbers
see mask3 below				No variable parts

NOTE: Mask1 below will be followed by either mask2 or mask3.

```
mask1 = 'Source log group <text> defined,'
mask2 = 'Log message numbers currently defined:'
       '        <text>'
mask3 = 'No log message numbers defined.'
```

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the log messages defined for a log group specified on the change_source_log_group command were successfully changed.

Log Message ID

CONST
    lsl$log_group_changed = min_log_message_id + 410;

MASK	LOG_MESSAGE_BUFFER		
fixed text	type	value	description
see mask1 below	char octet	1..31 octets	Name of the source log group changed
see mask2 below	bin_int	list of integer	List of log messages defined for log group
see mask3 below			No variable parts

NOTE: Mask1 below will be followed by either mask2 or mask3.

mask1 = 'Source log group <text> changed,'
mask2 = 'Log message numbers currently defined:'
       '   <text>'
mask3 = 'No log message numbers defined.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that a log group specified on the cancel_source_log_group command was successfully cancelled.

Log Message ID

CONST
    lsl$log_group_cancelled = min_log_message_id + 411;

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see mask1 below		char_ octet	1..31 octets	Name of log group cancelled

mask1 = 'Source log group <text> cancelled.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the LSA was not able to translate the Generic Transport title. The reason for the failure is indicated by the error code provided by Directory.

Log Message ID

```
CONST
   lsl$no_title_translate = min_log_message_id + 412;
```

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below		.	No variable parts

mask1 = 'The Log Support Application cannot obtain a Generic
        Transport title translation.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the LSA received a Generic Transport disconnect indication, and therefore is unable to log to the Independent Log M-E associated with the connection.

Log Message ID

CONST
    lsl$gt_disconnect_error = min_log_message_id + 413;

MASK	LOG_MESSAGE_BUFFER		
fixed text	type	value	description
see mask1 below			No variable parts
Reason code =	bin_ octet	1 octet	The reason code provided on disconnect indication

NOTE: The 'Reason code =' mask defined above is returned when a reason code is provided on the disconnect. If no reason code is provided this mask is not included.

mask1 = 'Disconnect indication received by the Log Support Application.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the LSA was not
able establish a connection via Generic Transport services.

Log Message ID

CONST
   lsl$gt_connect_error = min_log_message_id + 414;

MASK	LOG_MESSAGE_BUFFER		
fixed text	type	value	description
see mask1 below			No variable parts
Reason code =	bin_ octet	2 octets	The Generic Transport reason code provided

mask1 = 'The Log Support Application cannot establish a connection.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the LSA was not able transmit data over an established Generic Transport connection.

<u>Log Message ID</u>

CONST
    lsl$gt_data_request_error = min_log_message_id + 415;

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below			No variable parts
Reason code =	bin_ octet	2 octets	The Generic Transport reason code provided

mask1 = 'The Log Support Application cannot transmit via Generic Transport.'

6.0 LOG MESSAGES

This log message indicates that an LSA connection queue has
overflowed. Subsequent messages are discarded until the overflow
condition is resolved. The queues will overflow if a connection to
a log recorder cannot be established.

Log Message ID

CONST
    lsl$queue_overflow = min_log_message_id + 416;

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below			No variable parts

mask1 = 'LSA connection queue overflow, log messages are being
        discarded.'

6.0 LOG MESSAGES

This log message is generated by the Dependent Log M-E when it is
unable to abort a title translation using the Directory Services.

Log Message ID

```
CONST
   lsl$log_title_abort_error = min_log_message_id + 417;
```

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below	char octet	1..31 octets	Title name
see mask2 below	bin_ octet	2 octets	Directory error code

```
mask1 - 'Unable to abort translation for source log group title
         <text>,'
mask2 - 'Directory error: <text>.'
```

6.0 LOG MESSAGES

This log message is generated by the Dependent Alarm M-E when it is
unable to abort a title translation using the Directory Services.

<u>Log Message ID</u>

CONST
   ls1$alarm_title_abort_error = min_log_message_id + 418;

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below	char octet	1..31 octets	Title name
see mask2 below	bin_ octet	2 octets	Directory error code

mask1 - 'Unable to abort translation for source alarm group title
         <text>,'
mask2 - 'Directory error: <text>.'

6.0 LOG MESSAGES

This log message indicates that the LSA received a Generic Transport data indication. A data indication is not expected.

Log Message ID

```
CONST
    lsl$unexpected_data__indication =.min_log_message_id + 419;
```

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see maskl below		.	No variable parts

maskl = 'Unexpected data indication received by the Log Support
        Application.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that an LSA
connection queue overflow condition has been resolved. The number
of log messages that were discarded is included in the message.

Log Message ID

CONST
   lsl$log_messages_discarded = min_log_message_id + 420;

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see maskl below		bin_int	integer	The number of log messages discarded

maskl = 'Log overflow condition resolved, <text> log messages were
        discarded.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the LSA was not able to open a statistics SAP.

Log Message ID

```
CONST
  lsl$statistics_sap_not_opened = min_log_message_id + 421;
```

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below			No variable parts

mask1 = 'The Log Support Application could not open a statistics SAP.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that an unknown
intertask message workcode was received by the LSA. The DI will be
reset.

Log Message ID

CONST
   lsl$unknown_itm = min_log_message_id + 422;

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see mask1 below		bin_ octet	2 octets	The work code in the ITM

mask1 = 'Work code <text> unknown to the Log Support Application.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that an error was returned to the LSA on a Directory title translation request.

<u>Log Message ID</u>

CONST
    lsl$no_dir_title_translation = min_log_message_id + 423;

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see maskl below				No variable parts

maskl = 'The Log Support Application was not able to obtain a title
        translation.'

6.0 LOG MESSAGES

This log message reports the statistics collected by the LSA.

Log Message ID

CONST
   lsl$statistics_message = min_log_message_id + 424;

MASK		LOG_MESSAGE_BUFFER	
fixed text	type	value	description
see maskl below	bin_int	integer	The number of log/alarm messages received
	bin_int	integer	The number of log messages transmitted
	bin_int	integer	The number of alarm messages transmitted
	bin_int	integer	The number of disabled messages not transmitted
	bin_int	integer	The number of log messages discarded
	bin_int	integer	The number of alarm messages discarded
	bin_int	integer	The number of messages discarded due to no buffers

```
maskl =
'LOG_SUPPORT_APPLICATION Statistics from <n> to <n> reason = <n>
 group = <n>'
'     log/alarm messages received = <n>'
'     defined log messages transmitted = <n>'
'     defined alarm messages transmitted = <n>'
'     disabled log/alarm messages discarded = <n>'
'     log messages discarded due to queue limit = <n>'
'     alarm messages discarded due to queue limit = <n>'
'     log/alarm messages discarded due to buffer/memory congestion
      = <n>'
```

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the alarm group
specified on the define_source_alarm_group command was defined.

Log Message ID

CONST
    lsl$alarm_group_defined = min_log_message_id + 425;

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see maskl below		char octet	1..31 octets	Source alarm group name

maskl = 'Source alarm group <text> defined.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that a change has been made to the set of alarm message numbers defined, by either the define_source_alarm_message or cancel_source_alarm_message command.

Log Message ID

```
CONST
    lsl$alarm_messages_defined = min_log_message_id + 426;
```

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see mask1 below		bin_int	integer	List of defined message numbers

mask1 = 'Alarm message numbers currently defined:'
       '   <text>'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that all alarm
messages have been cancelled by the cancel_source_alarm_message
command.

<u>Log Message ID</u>

CONST
   lsl$alarm_messages_cancelled = min_log_message_id + 427;

MASK		LOG_MESSAGE_BUFFER		
fixed text		type	value	description
see mask1 below				No variable parts

mask1 = 'All alarm message numbers cancelled.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that the alarm group
specified on the cancel_source_alarm_group command was deleted.

Log Message ID

CONST
    lsl$alarm_group_cancelled = min_log_message_id + 428;

M A S K	L O G _ M E S S A G E _ B U F F E R		
fixed text	type	value	description
see mask1 below	char_ octet	1..31 octets	Source alarm group name

mask1 = 'Source alarm group <text> cancelled.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that an LSA alarm connection queue overflow condition has been resolved and to indicate the number of alarm messages that were discarded.

Log Message ID

```
CONST
    lsl$alarm_messages_discarded = min_log_message_id + 429;
```

MASK	LOG_MESSAGE_BUFFER		
fixed text	type	value	description
see mask1 below	bin_int	integer	The number of alarm messages discarded

mask1 = 'Alarm overflow condition resolved, <text> alarm messages were discarded.'

6.0 LOG MESSAGES

The purpose of this log message is to indicate that a disconnect indication was issued by the Independent Alarm M-E.

<u>Log Message ID</u>

CONST
    lsl$unexpected_alarm_disconnect = min_log_message_id + 430;

M A S K		L O G _ M E S S A G E _ B U F F E R		
fixed text		type	value	description
see maskl below				No variable parts

maskl = 'Disconnect indication received by the Dependent Alarm M-E.'

7.0 NEW DATA TYPES

7.0 <u>NEW DATA TYPES</u>


```
  CONST
    min_log_message_id = 0,
    max_log_message_id = 32999;

  TYPE
    log_msg_id_type = min_log_message_id .. max_log_message_id;

  TYPE
    log_priority = (log_critical, log_high, log_medium, log_low);


{ CDCNET log message map.

  CONST
    number_of_long_words = (max_log_message_id - min_log_message_id + 31)
      DIV 32;

  TYPE
    long_words = 1 .. number_of_long_words;

  TYPE
    lsa_message_map_type = packed array [min_log_message_id ..
      max_log_message_id] of boolean;

  TYPE
    message_map_type = record
      case boolean of
      = TRUE =
        map: lsa_message_map_type,
      = FALSE =
        init: array [long_words] of integer,
      casend,
    recend;


{ CDCNET system log message vector.

  VAR
    log_message_vector: [XDCL, STATIC] lsa_message_map_type := [REP (max_log_messag
      min_log_message_id + 1) OF TRUE];


{ Logging / Alarming queue sizes.

  VAR
```

CONTROL DATA PRIVATE

7.0 NEW DATA TYPES

```
    connection_queue_size: [XDCL] 0 .. 0ffff(16) := 250,
    max_preserve_queue_size: [XDCL] 0 .. 0ffff(16) := 15;


{ LSA task attributes.

  VAR
    log_support_application_ta: [XDCL, READ] task_attributes := [
      2048,    { stack size
      3,       { priority
      FALSE,   { not premptable
      FALSE];  { not immediate control
```