

DISTRIBUTED COMMUNICATIONS NETWORK
(CDCNET)

APPENDIX A -- CDCNET MESSAGE TEMPLATES

R. R. Rundquist
05/13/85

DISCLAIMER: This document is an internal working paper only. It is unapproved and subject to change, and does not necessarily represent any official intent on the part of Control Data Corporation.

REVISION RECORD			
Revision	Description	Author	Date
A	Original version	RRR	02/27/85
B	Updated per preliminary review	RRR	04/13/85
C	Updated per TDRB review	RRR	05/13/85

05/13/85

Table of Contents

A1.0 INTRODUCTION	A1-1
A1.1 REFERENCES	A1-1
A2.0 CDCNET MESSAGES	A2-1
A2.1 MESSAGE TEMPLATE DEFINITION FORMAT	A2-1
A2.1.1 SEVERITY LEVELS	A2-1
A2.1.1.1 CDCNET ALARM MESSAGE SEVERITY LEVEL DEFINITIONS	A2-2
A2.1.1.2 CDCNET COMMAND RESPONSE SEVERITY LEVEL DEFINITIONS	A2-3
A2.1.2 MESSAGE TEMPLATE TEXT	A2-4
A2.1.3 MESSAGE PARAMETER TEXT	A2-7
A2.1.4 EXAMPLE TEMPLATE DECLARATIONS	A2-9
A2.1.4.1 CDCNET LOG MESSAGE USING ONE MESSAGE TEMPLATE	A2-9
A2.1.4.2 CDCNET LOG MESSAGE USING TWO MESSAGE TEMPLATES	A2-9
A2.1.4.3 CDCNET COMMAND RESPONSE USING ONE MESSAGE TEMPLATE	A2-10
A2.1.5 TEMPLATE DEFINITION COMMON DECKS	A2-11
A3.0 SITE CONFIGURABLE OPTIONS	A3-1
A3.1 BINARY RELEASE	A3-1
A3.2 SOURCE CODE RELEASE	A3-1
A3.2.1 CHANGING EXISTING TEMPLATES	A3-1
A3.2.2 ADDING NEW TEMPLATES	A3-2
A3.2.3 DELETING EXISTING TEMPLATES	A3-2
A4.0 NATURAL LANGUAGE AND MESSAGE TEMPLATES	A4-1
A5.0 CREATING MESSAGE TEMPLATES	A5-1
A5.1 GENERATE_MESSAGE_TEMPLATES GENMT	A5-1
A5.2 CREATE_MESSAGE_MODULE CREMM	A5-3
A5.3 NETWORK_INITIALIZE_TEMPLATE_FILE / NETITF	A5-3
A6.0 CYBER 170/NOS MESSAGE TEMPLATE SUPPORT FOR CDCNET	A6-1
A6.1 NETWORK PERFORMANCE ANALYZER (NPA)	A6-1
A6.1.1 GENERATING A CDCNET MESSAGE TEMPLATE DATA STORE FOR NPA	A6-1
A6.1.2 CDCNET NETWORK PERFORMANCE REPORTS	A6-2
A6.2 NETWORK OPERATOR SERVER (NETOS)	A6-2
A6.2.1 GENERATING A CDCNET MESSAGE TEMPLATE DATA STORE FOR NETOS	A6-2
A6.2.2 CDCNET ALARMS	A6-3
A6.2.3 CDCNET NETWORK COMMAND RESPONSES	A6-4

05/13/85

A1.0 INTRODUCTION

A1.0 INTRODUCTION

Message templates are used within CDCNET to facilitate meaningful displays for human interface. Templates are used to format command responses, alarm messages, and network performance reports that are reported to a network operator or terminal user.

This appendix describes how message templates are used within CDCNET, how the templates are defined, site configurable options, and how message template data stores are created.

Message templates contain data which remains constant from one instance or execution to the next. In other words text within a display message which remains constant from one report to the next is placed within message templates. CDCNET transfers only the data which changes from one instance to the next (i.e., the variable data) within the network. The variable data gets combined with the appropriate message template to form a human readable display.

Message templates are defined such that the templates can be tailored without recompiling the procedure which produces the corresponding variable data.

The document entitled 'CDCNET Network Operator Message Conventions' contains the conventions to be used when generating messages to be output from the CDCNET network. This appendix covers the syntax and definition of CDCNET messages as opposed to the semantics of a message.

A1.1 REFERENCES

Software Design Guide (CDC-PUB 15011400) July 84

CDCNET Network Operator Message Conventions (ARH6743)- R. Rundquist

CYBIL for NOS/VE System Interface (Pub #60464115)

NOS/VE System Command Language Command Writer's Guide

NOS/VE ERS - Program Interface (ARH3619)

05/13/85

A2.0 CDCNET MESSAGES

A2.0 CDCNET MESSAGES

A single human readable display message is constructed via the use of one or more CDCNET message templates. The first display line pertinent to the specific condition being reported, will contain a description of the condition and the severity level of the condition being reported. The initial lines of a display will always be a CDCNET header. Below is an example of a CDCNET alarm message. The first two lines contain the CDCNET alarm message header while the third and fourth lines contain the specific information pertinent to the condition being reported.

```
CDCNET ALARM *****
system_name 83/08/04 11.00.35 30432
--ERROR-- Incorrectly formatted Routing Information Data Unit
Routing Information Data Unit = ffedcl2450cdcd120123ccf
```

A2.1 MESSAGE TEMPLATE DEFINITION FORMAT

A message template declaration consists of a CYBIL constant declaration (i.e., template id) followed by a CYBIL comment field which contains the severity level and the message template text. The constant declaration equates a template identifier to an integer template id.

Message definitions are terminated by a blank line.

```
template id;
{<severity> <message text (upto 80 characters)>}
{<message text continued (if necessary) [upto 80 characters]>}
```

A2.1.1 SEVERITY LEVELS

The severity level for the message applies to the severity of the condition.

Below is list of the SEVERITY LEVELs which can be assigned to a message, and the associated codes to be used:

- 'I' INFORMATIVE CONDITION
- 'W' WARNING CONDITION
- 'E' ERROR CONDITION

05/13/85

A2.0 CDCNET MESSAGES
A2.1.1 SEVERITY LEVELS

'F' FATAL CONDITION

'C' CATASTROPHIC CONDITION

If a message is constructed with the use of 2 or more message templates the first template (which contains the condition description), will be defined with the severity level which will apply to the entire message. Additional templates used for forming the message should be defined with a severity level of INFORMATIVE.

A2.1.1.1 CDCNET ALARM MESSAGE SEVERITY LEVEL DEFINITIONS

<u>SEVERITY LEVEL</u>	<u>DESCRIPTION</u>
<u>INFORMATIVE CONDITION</u>	These messages convey items of general interest and are not a result of incorrect or incomplete operation. Used for Statistical type messages, system event messages (e.g., logging of operator activity -- including all commands entered, responses received, and alarms received), periodic reporting of system configuration information, accounting type information, etc.
<u>WARNING CONDITION</u>	These messages convey items of general interest and may have been the result of incorrect or incomplete operation. Warning indicates that the system is approaching some error condition (i.e., threshold condition). Used for messages containing information warning of system resource degradation (e.g., used by Executive to report availability of buffers, memory, etc.; has degraded below the acceptable threshold), etc.
<u>ERROR CONDITION</u>	These messages convey that the operation was not completed correctly. A message is qualified as severity level error if the message is the result of an error condition which is correctable by the DI software (i.e., error has minimal impact on system operation and performance). Used for parameter verification errors (e.g., if a communication layer or network management entity receives invalid parameters at its

05/13/85

A2.0 CDCNET MESSAGESA2.1.1.1 CDCNET ALARM MESSAGE SEVERITY LEVEL DEFINITIONS

user interface, the error is logged by the layer and the user is notified of the error). This level should be used when reporting the reception of bad PDUs. This severity level should also be used in the case that an action is not allowed at the time of request.

FATAL CONDITION

These messages convey that the operation was not completed correctly. A message is qualified as severity level fatal if the message is the result of an error condition which affects the operation of a major portion of the system but was recoverable by the DI software. Used to indicate fatal hardware failures (e.g., used by lower layer failure management for device failures), and fatal software conditions (e.g., used by System Ancestor to report the occurrence of a task failure).

CATASTROPHIC CONDITION

These messages convey that the operation did not complete correctly and resulted in the least desired recovery. A message is qualified as severity level catastrophic if the message is the result of an error condition which has severe impact on system/network operation and performance. Used for severe hardware failures which affect a large portion of the system (used by lower layer failure management to report severe device failures). Used by System Ancestor to indicate that the system required reloading due to numerous task failures. Catastrophic means that the DI could not recover without reload.

A2.1.1.2 CDCNET COMMAND RESPONSE SEVERITY LEVEL DEFINITIONSSEVERITY LEVELDESCRIPTIONINFORMATIVE CONDITION

These messages convey items of general interest and are not a result of incorrect or incomplete operation. This severity level should be used to indicate a normal response. Used by Network Command Processors.

05/13/85

A2.0 CDCNET MESSAGESA2.1.1.2 CDCNET COMMAND RESPONSE SEVERITY LEVEL DEFINITIONS

WARNING CONDITION

These messages convey items of general interest and may have been the result of incorrect or incomplete operation. Used by Network Command Processors.

ERROR CONDITION

These messages convey that the operation was not completed correctly. This severity level will be used for parser errors, parameter verification errors, and in cases where the action to be performed is not allowed at the time of request. Used by OSA, Network Command Processors and SCL Parser.

FATAL CONDITION

These messages convey that the operation was not completed correctly. This level should be used where an action can not be completed due to device failures or where an important action can not be performed but does not cause adverse affects on the system operation. Used by OSA and Network Command Processors. For example this severity level will be used by OSA when reporting that a command could not be delivered to the destination system.

CATASTROPHIC CONDITION

These messages convey that the operation did not complete correctly. This severity level should be used in the following cases; Command Processor Task aborted, no response received from destination system, OSA Transport connection failure to destination system (i.e., Dependent Command M-E), etc. Used by Dependent Command M-E and OSA.

A2.1.2 MESSAGE TEMPLATE TEXT

A message template is a string of text characters and control sequences. Each text character in the template is copied to the output text when a message is formatted.

The plus character (+) is used to introduce a control character within a template. This control character and the characters immediately following it designate actions the message formatter is to perform with respect to the output text. In the following descriptions of the control sequences, the notation [n] is used to indicate that n is optional. n is defined as an unsigned decimal

05/13/85

A2.0 CDCNET MESSAGESA2.1.2 MESSAGE TEMPLATE TEXT

number between 1 and 128.

The control sequences are:

+P[n] The nth variable field is inserted in the output text. If n is omitted, n is assumed to be one more than the value used in the previous call. If there has not been a previous call, the first variable field is inserted.

Assuming the following variable field values:

'111', '222', '333', '444',

'abc+Pdef+P3ghi+Pjkl+P2' will cause
"abc111def333ghi444jkl222" to appear in the output text.

'abc+P-123' will cause
"abc111-123" to appear in the output text.

+N[n] Start a new line in the output text indented by n spaces. If n is omitted, the new line is not indented.

'now+N2is+N4the+Ntime' will cause

now
 is
 the
 time"

to appear in the output text.

Note that a '+N' must be explicitly specified in the template definition for a new line to occur (i.e., no new lines are generated implicitly).

+X[n] Insert n spaces in the output text. If n is omitted, 1 space is inserted.

'abc+X5def' will cause "abc def" to appear in the output text.

++ Insert a single "+" in the output text.

'abcd++efg' will cause "abcd+efg" to appear in the generated text.

+ - This option inserts nothing in the output text. It is used to separate control characters from text characters where there can be a conflict.

05/13/85

A2.0 CDCNET MESSAGES

A2.1.2 MESSAGE TEMPLATE TEXT

'abc+Xl+-lxx' will cause "abc lxx" to appear in the generated text, where as 'abc+Xllxx' will cause "abc xx" to appear in the generated text.

+H[n] Insert spaces until column n of the current line in the output text is reached. If the current position in the output text is past column n, then one space will be inserted. If n is omitted then insert spaces until the next default tab position in the current line in the output text is reached. Default tab positions occur every 8 columns.

'abc+H10efg+H20hij+H21' will cause
"abc efg hij"

+R Begin repeating information. The rest of the message text is assumed repeated indefinitely until all the delimited sequences from the text field of the status record are exhausted. NOTE: If n is specified for a P control sequence within the repeating information, the information will be repeated an infinite number of times (i.e., infinite loop -- be careful not to have n specified on the P parameter).

Assume the following variable field values:
'four', 'score', 'and', 'seven', 'years', 'ago'

'+P+R+H+P' will produce:
"four score and seven years ago"

'+R+P+H+P+H+P+N' will produce:
"four score and
seven years ago"

Alphabetic control characters may appear in the template in either upper or lower case.

If the character following the control character is not one of those quoted above, the message formatter will not take any action but outputs the characters verbatim.

For a single template definition the output display message is limited to 24 display lines, each line being limited to 72 characters. The template definition itself is limited to 80 characters per 'message text' line.

05/13/85

A2.0 CDCNET MESSAGESA2.1.3 MESSAGE PARAMETER TEXT

A2.1.3 MESSAGE PARAMETER TEXT

Parameters for insertion into formatted messages are obtained from the text field of the status record (i.e., the variable data fields generated from within the DI). A status parameter is an item of information inserted into the message template to make the message meaningful. For example, it could be the name of the command for which the error was detected or it could be a string whose syntax is incorrect. The procedure `gen_data_field` is used within the DI to generate the variable data fields for a message. The data fields are generated in Management Data Unit format (MDU). Refer to section 7.0 entitled 'Management' of the CDNA General Design Specification for further details on MDU. Section 7.0 describes the format of the data when represented in MDU format.

05/13/85

A2.0 CDCNET MESSAGES

A2.1.3 MESSAGE PARAMETER TEXT

The allowed data types for the variable data elements are:

<u>DATA TYPE</u>	<u>DESCRIPTION</u>
<u>Binary Octet (bin octet)</u>	Binary octets consist of bytes of binary information. The length field passed to <code>gen_data_field</code> indicates the number of bytes. Binary octets are displayed as their equivalent hexadecimal value. A single octet becomes a 2 digit hex ASCII display.
<u>Character String (char octet)</u>	Character string data contains ASCII characters. The length passed to <code>gen_data_field</code> indicates the numbers of ASCII characters.
<u>Binary String (bin str)</u>	A binary string is a contiguous string of bits. The length passed to <code>gen_data_field</code> indicates the number of bits. A binary string is displayed as 0's and 1's.
<u>Unsigned Integer (bin int)</u>	An integer contains a string of binary bits. The length passed to <code>gen_data_field</code> indicates the number of bits in the integer. A data element of type integer gets displayed as its equivalent decimal value.
<u>Signed Integer (bin sint)</u>	An integer contains a string of binary bits. The length passed to <code>gen_data_field</code> indicates the number of bits in the integer. A data element of type integer gets displayed as its equivalent decimal value. The most significant bit contains the sign bit.
<u>Binary Coded Dec (bcd char)</u>	BCD values range from 0 to 9 or 0000(2) to 1001(2) respectively. Each octet contains 2 BCD data elements. The length field passed to <code>gen_data_field</code> indicates the number of BCD elements in the field. A data element of type BCD gets displayed as its equivalent decimal value.

05/13/85

A2.0 CDCNET MESSAGESA2.1.4 EXAMPLE TEMPLATE DECLARATIONS

A2.1.4 EXAMPLE TEMPLATE DECLARATIONS

The message template defines the template id, severity level, and message template text to be used for the message. Note that the 3rd and 4th characters of the template id must be 'e\$'.

```
template id;
{<severity> <message text (upto 80 characters)>}
{<message text continued (if necessary) [upto 80 characters]>}
```

A2.1.4.1 CDCNET LOG MESSAGE USING ONE MESSAGE TEMPLATE

Below is an example of a CDCNET log message template definition. The message is reporting that the Routing Management Entity has received an incorrectly formatted Routing Information Data Unit.

CONST

```
rme$rme_bad_ridu = cme$min_template_id + 21;
{E Incorrectly formatted Routing Information Data Unit received}
{+NRouting Information Data Unit = +P1}
```

O P E R A T O R D I S P L A Y E X A M P L E

This shows how the message would appear if it were displayed as an alarm at the operators console.

```
CDCNET ALARM *****
system_name 83/08/04 11.00.35 30432
--ERROR-- Incorrectly formatted Routing Information Data Unit received
Routing Information Data Unit = ffedcl2450cdcd120123ccf
```

A2.1.4.2 CDCNET LOG MESSAGE USING TWO MESSAGE TEMPLATES

This example shows the message templates Routing Management uses to report the reception of a Routing Information Data Unit containing an invalid network address.

05/13/85

A2.0 CDCNET MESSAGES

A2.1.4.3 CDCNET COMMAND RESPONSE USING ONE MESSAGE TEMPLATE

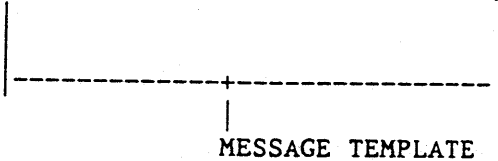
operators console.

```
system_name 83/08/04 11.00.35 4033
--FATAL-- Command could not be delivered to system arh_101
```

This example shows the message template used by the CDCNET DISPLAY_MEMORY command to report a memory display.

CONST

```
exe$display_memory_template = cme$min_template_id + 32;
{I display_memory+N}
{+R+P+H12+P+H22+P+H32+P+H42+P+N}
```



O P E R A T O R D I S P L A Y E X A M P L E

This shows how the message would appear when displayed at an operators console.

```
system_name 83/08/04 11.00.35 1232
--INFORMATIVE-- display_memory
0000f3e2    ffed    4e34    defd    de45
0000f3ea    deda    12ed    345f    ed43
```

A2.1.5 TEMPLATE DEFINITION COMMON DECKS

In order to be able to produce a listing containing all of the template definitions that CDCNET can generate, every functional area is required to provide a common deck which contains all of the area's message template definitions.

This common deck should be named XXETEMP, where XX is the functional area's "group code" as defined by CDCNET integration.

These common decks will be called into a deck named CMCTEMP which can be compiled to produce a listing containing all of the message templates currently defined in the system.

The deck METMDU on the CDCNET Source Program Library should be called to obtain the CYBIL definition of the MDU data element types.

05/13/85

A3.0 SITE CONFIGURABLE OPTIONS

A3.0 SITE CONFIGURABLE OPTIONS

A3.1 BINARY RELEASE

With a binary release of CDCNET, message templates will not be changeable nor will additions or deletions to the existing templates be allowed. A customer must purchase CDCNET source code in order to take advantage of the configurable options.

A3.2 SOURCE CODE RELEASE

With source code the customer can change message templates. A customer may:

- change existing templates.
- add new templates to support new messages being generated.
- delete existing templates.

A3.2.1 CHANGING EXISTING TEMPLATES

A customer may choose to change the message text of the template and/or the severity level. The customer changes the CDCNET network template definitions by editing the CDCNET source file (i.e., 'ETEMP' deck) which contains the template(s) to be changed. Any text editor can be used to change the file.

The CDCNET deck 'CMMTEMP' must then be run through GENCOMP (GENERate COMPile file) to generate a file which can then be run through the Generate Message Template utility to create a Cybil source module containing the message templates. The Cybil module must then be compiled to produce a new loadable object module containing the message templates. This produced template data store is for use by the Network Performance Analyzer (NPA) in producing network performance reports.

The output of the GENCOMP specified above must also be run through FCOPY to convert it to 8/8 ASCII. This 8/8 ASCII file must then be run through the NETWORK Initialize Template File utility (NETITF) to generate a Binary Template Definition File that NETOS will use at execution time. NETOS uses this data store when formatting CDCNET

05/13/85

A3.0 SITE CONFIGURABLE OPTIONS

A3.2.1 CHANGING EXISTING TEMPLATES

command responses and alarm messages for display. See sections 5.0 and 6.0 for more details on creating message templates.

A3.2.2 ADDING NEW TEMPLATES

New templates will be added by customers doing customer developed software for the DI. All CDC CDCNET template ids must be in the range 1 to 9999. Template ids in the range of 10000 to 19999 are reserved for message templates for customer developed software. The remainder of the template id range is reserved for future use (i.e., 20000 to 65535). CDCNET adheres to the template id guidelines defined in the NOS/VE Program Interface ERS section 2.10.4.

The customer adds new template definitions by editing an existing CDCNET source file (i.e., 'ETEMP' deck) for the existing area in which new templates are to be added, or by creating new 'ETEMP' decks. Any new 'ETEMP' decks must be added to 'CMCTEMP'. Any text editor can be used to change the file. The CDCNET deck 'CMMTEMP' must then be run through GENCOMP (GENERate COMPile file) to generate a file which can then be run through the Generate Message Template utility to create a Cybil source module containing the message templates. The Cybil module must then be compiled to produce a new loadable object module containing the message templates, for use by NPA.

The output of the GENCOMP specified above must also be run through FCOPY to convert it to 8/8 ASCII. This 8/8 ASCII file must then be run through the NETWORK Initialize Template File utility (NETITF) to generate a Binary Template Definition File that NETOS will use at execution time. Section 5.0 and 6.0 contain more information on the generation of the message template data stores including the specific commands.

A3.2.3 DELETING EXISTING TEMPLATES

If an existing template is deleted, the default provided message template will be used. Any time the message formatter searches the template data store and can not find the template corresponding to the specified template id, the default message template will be used. The CDC provided default message template is shown below:

```
ID=xx CC=code TEXT=string
```

where ID is the CDCNET product code identifier, CC is the template id, and TEXT is an ASCII dump of the variable data fields supplied

05/13/85

A3.0 SITE CONFIGURABLE OPTIONS**A3.2.3 DELETING EXISTING TEMPLATES**

for the message template. The default message template will be used for each non-existent template used in a message.

The customer deletes template definitions by editing the CDCNET source file (i.e., 'ETEMP' deck) which contains the template(s) to be deleted. Any text editor can be used to change the file. The CDCNET deck 'CMMTEMP' must then be run through GENCOMP (GENERate COMPile file) to generate a file which can then be run through the Generate Message Template utility to create a Cybil source module containing the message templates. The Cybil module must then be compiled to produce a new loadable object module containing the message templates, for use by NPA.

The output of the GENCOMP specified above must also be run through FCOPY to convert it to 8/8 ASCII. This 8/8 ASCII file must then be run through the NETwork Initialize Template File utility (NETITF) to generate a Binary Template Definition File that NETOS will use at execution time. Refer to section 5.0 and 6.0 for more details.

05/13/85

A4.0 NATURAL LANGUAGE AND MESSAGE TEMPLATES

A4.0 NATURAL LANGUAGE AND MESSAGE TEMPLATES

CDCNET supports the simultaneous existence of a number of versions of message templates. Each version is written in a different natural language (e.g., American, Danish, Dutch, English, Finnish, Flemish, French, German, Italian, Norwegian, Portugese, Spanish, Swedish).

A separate set of 'ETEMP' decks will be supported by CDCNET for each natural language. Only American will supported for CDCNET Release 1.0.

All message templates in a given message module are assumed to be written in the same natural language, and the name of that language is specified on the `CREATE_MESSAGE_MODULE` command (the default natural language is American).

For Release 1.0 of CDCNET, the `CREATE_MESSAGE_MODULE` command will not be required. CDCNET will use an existing version of GENMT which runs on the Cyber 170/NOS to generate and maintain the message template data stores. This version of GENMT differs from the version which runs on the Cyber 180/NOSVE in that the `CREATE_MESSAGE_MODULE` step is not required. On the Cyber 180/NOSVE, GENMT takes the input (which is the same as the Cyber 170/NOS GENMT input) and forms commands from the defined message templates. The `CREATE_MESSAGE_MODULE` command is then invoked to generate the actual template data store (the output is the same as the output generated by the Cyber 170/NOS GENMT). Refer to section 5.2 for more detail on the `CREATE_MESSAGE_MODULE` command. `CREATE_MESSAGE_MODULE` facilitates the support for multiple natural languages.

`CREATE_MESSAGE_MODULE` will be used when CDCNET message templates are supported on the Cyber 180/NOSVE (Release 2.0).

05/13/85

A5.0 CREATING MESSAGE TEMPLATES

A5.0 CREATING MESSAGE TEMPLATES

The descriptive information for a message template resides in a "message module". The descriptive information consists of the name of the message template, template id, its severity level, and the message template text.

A message module is built using the CREATE_MESSAGE_MODULE subutility and GENERATE_MESSAGE_TEMPLATES utility. For Release 1.0 CREATE_MESSAGE_MODULE is not required.

A5.1 GENERATE MESSAGE TEMPLATES | GENMT

Generate_message_templates is a tool which provides for the generation of a message template module from a set of input templates. This is accomplished by supplying as input to this tool, the definitions of the CYBIL constant identifiers for template ids along with CYBIL comments which are interpreted to represent the severity level and message template text.

The output produced by generate_message_templates is a file containing the message template information in the form of a "nonexpanded" CYBIL module. This output can then be expanded and compiled to produce a template object module for use in execution.

```
generate_message_templates input=<file>
                             [output = <file>]
                             [error = <file>]
                             [product_identifier = <name>]
                             [status = <status_variable>]
```

input|i : This parameter specifies the file containing the message templates. See the next section for information and examples on how to construct these templates.

output|o : This parameter specifies the file to contain the generated message templates. This file can be compiled to produce an object module of templates for use.

Omission of this parameter will result in the value of COMPILE being used.

error|e : This parameter specifies the file to report any errors encountered during template generation.

05/13/85

A5.0 CREATING MESSAGE TEMPLATES
A5.1 GENERATE_MESSAGE_TEMPLATES | GENMT

Omission of this parameter will result in the value of \$ERRORS being used.

product_identifier|pi : This parameter specifies the two character identifier to be used in constructing the template module name.

Omission of this parameter will result in the value of OS being used.

status : See ERROR HANDLING in the NOS/VE Command Interface ERS.

The format and processing of the input file is as follows:

- Any line beginning with a CYBIL name where the third and fourth characters are e\$ is considered a constant to be evaluated. An e\$ constant is considered to be the start of a template definition.

- Only one constant declaration per line is recognized; and the constant declaration must be on one line.

* If an e\$ constant is declared more than once the first definition definition is used.

- Any consecutive lines beginning with '{' directly following an e\$ constant declaration are processed for the error severity level and message text. The lines may contain a corresponding '}' to signify the end of the line, or the '}' may be omitted.

* The severity level indicator is the first character after '{' of the first line. A character is recognized as a severity indicator if it is the first letter (either upper or lower case) of the name of the severity level (i.e., 'I', 'W', 'E', 'F', or 'C'). If one is not found catastrophic ('C') is assumed.

- An elipsis may be optionally included at the end of a line of message text if the text is continued onto the next line.

* The message text is truncated if it exceeds the maximum size allowed.

- All lines other than those mentioned above and excess characters of processed lines are ignored. For example, the following line would be ignored:

```
ignore_constant = 1;
```

and in the following line:

05/13/85

A5.0 CREATING MESSAGE TEMPLATES

A5.1 GENERATE_MESSAGE_TEMPLATES | GENMT

xxe\$error = cme\$min_template_id + 10; {E message text

the characters starting with ';' through to the end of the line are ignored.

- Message template definitions are terminated by a blank line.

Error messages are issued for all errors that may be found while processing a line. At such points the e\$ constant declaration or complete condition definition is ignored except for those situations noted above with '*'.

A5.2 CREATE_MESSAGE_MODULE | CREMM

On the Cyber 180s/NOSVE a message module is built using the CREATE_MESSAGE_MODULE subutility of the CREATE_OBJECT_LIBRARY utility. For the CYBIL programmers, the GENERATE_MESSAGE_TEMPLATES command can be used to "front end" message module creation so that the production of the corresponding descriptive information is more readily coordinated. For further detail on the CREATE_MESSAGE_MODULE subutility refer to section 3.0 of the document entitled 'Object Code Management for NOS/VE' (Pub Number 60464413).

The CREATE_MESSAGE_MODULE subutility is not required for supporting message templates on the Cyber 170s/NOS.

A5.3 NETWORK_INITIALIZE_TEMPLATE_FILE / NETITF

NETITF will only be required for support of CDCNET message templates on the Cyber 170/NOS. The Cyber 180/NOSVE will use GENMT and CREATE_MESSAGE_MODULE for generating the template data store for the Cyber 180/NOSVE CDCNET network operator interface.

The template definition file read by the Cyber 170/NOS CDCNET Network Operator Server (NETOS) must have been produced by the template file preprocessor NETITF. This utility is similar to the SES routine GENMT, except that instead of producing a CYBIL template module, it produces a special file that is read in by NETOS whenever NETOS is initialized. This file is a binary template definition file which is then used by NETOS for its message formatting.

netitf base = <integer>

05/13/85

A5.0 CREATING MESSAGE TEMPLATESA5.3 NETWORK_INITIALIZE_TEMPLATE_FILE / NETITF

[t = <file>]

[b = <file>]

- base** : This parameter specifies the base template number. This number should be one less than the lowest template number in the template definition file. The default template definition will be assigned this value. This default template is defined as '+R+X+P'. The default message template will be used by NETOS for each non-existent template used in a message.
- t** : This parameter specifies the file name of the source file containing the template definitions.
Omission of this parameter will result in the value of `TEMPLAT`.
- b** : This parameter is the local file name of the binary template definition file written by NETITF. This file will be read by NETOS.
Omission of this parameter will result in the value of `TEMPLTB`.

05/13/85

A6.0 CYBER 170/NOS MESSAGE TEMPLATE SUPPORT FOR CDCNET

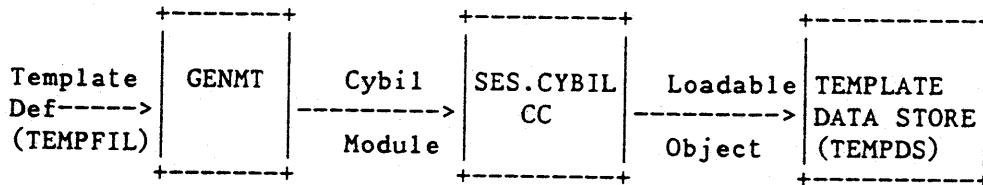
A6.0 CYBER 170/NOS MESSAGE TEMPLATE SUPPORT FOR CDCNET

A6.1 NETWORK PERFORMANCE ANALYZER (NPA)

A6.1.1 GENERATING A CDCNET MESSAGE TEMPLATE DATA STORE FOR NPA

SES,CDNABLD.GENCOMP M=CMMTEMP,B=PL10XX,UN=CDNA,CF=TEMPFIL.
where XX is the current build level
SES,CDNABLD.GENMT I=TEMPFIL,PI=DC.
SES,CDNABLD.CYBIL I=TEMPLAT,B=TEMPDS,CC,L=LISTING.

TEMPDS contains a loadable object module containing
the message template data store.

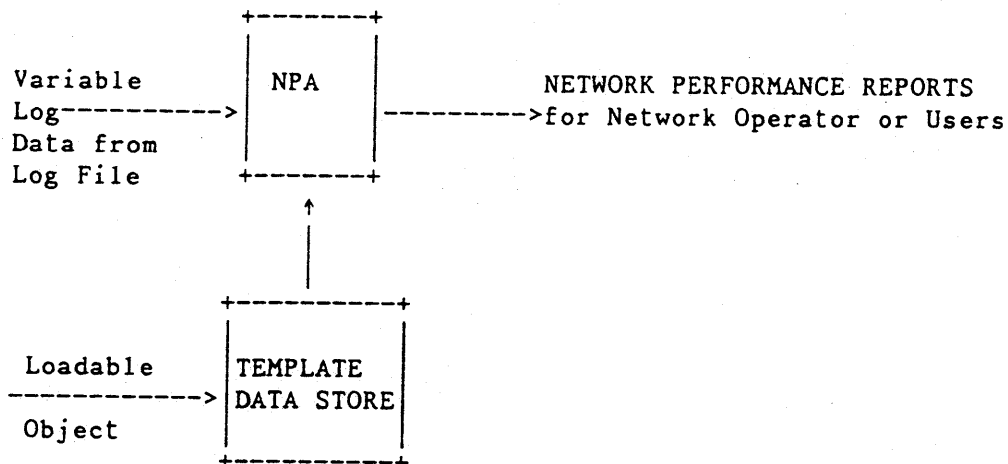


The NPA template data store is linked into the NPA field space
at NPA build time.

05/13/85

 A6.0 CYBER 170/NOS MESSAGE TEMPLATE SUPPORT FOR CDCNET
 A6.1.2 CDCNET NETWORK PERFORMANCE REPORTS

A6.1.2 CDCNET NETWORK PERFORMANCE REPORTS

A6.2 NETWORK OPERATOR SERVER (NETOS)

A6.2.1 GENERATING A CDCNET MESSAGE TEMPLATE DATA STORE FOR NETOS

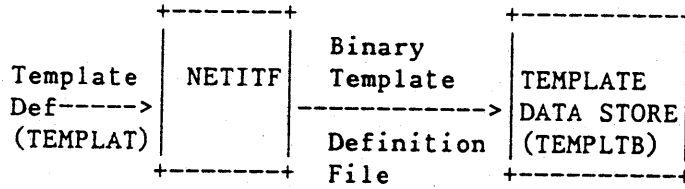
```

SES,CDNABLD.GENCOMP M=CMMTEMP,B=PL10XX,UN=CDNA,CF=TEMPFIL.
  where XX is the current build level
FCOPY,P=TEMPFIL,N=TEMPLAT,PC=ASCII,NC=ASCII88,R.
RFL,220000.
NETITF,BASE=0,T=TEMPLAT,B=TEMPLTB.
  TEMPLTB is the CDCNET message template file which will
  be read by NETOS at run time.
  
```

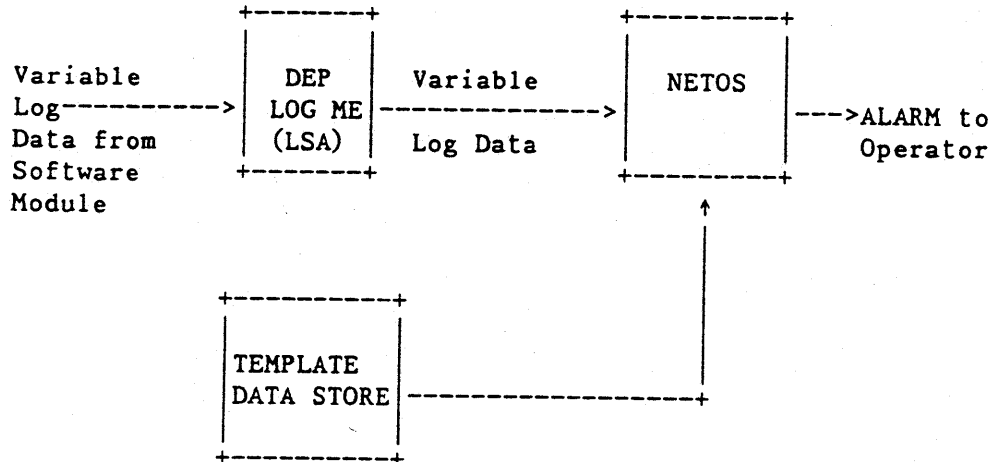
05/13/85

A6.0 CYBER 170/NOS MESSAGE TEMPLATE SUPPORT FOR CDCNET

A6.2.1 GENERATING A CDCNET MESSAGE TEMPLATE DATA STORE FOR NETOS



A6.2.2 CDCNET ALARMS



05/13/85

A6.0 CYBER 170/NOS MESSAGE TEMPLATE SUPPORT FOR CDCNET
A6.2.3 CDCNET NETWORK COMMAND RESPONSES

A6.2.3 CDCNET NETWORK COMMAND RESPONSES

