TAM-1001-017

DMS-100 Family

Automatic Message Accounting (AMA) Tools User Guide

Technical Assistance Manual

BCS35 and up   Standard 01.01   March 1993

northern
telecom

DMS-100 Family

Automatic Message Accounting (AMA) Tools User Guide

Technical Assistance Manual

Publication number: TAM-1001-017
Product release: BCS35 and up
Document release: Standard 01.01
Date: March 1993

Printed in the United States of America

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

Publication history

March 1993

BCS35 Standard 01.01 first release of this document

Contents

List of figures

List of tables

About this document

The AMA Tools document supports the software diagnostic tools AMADUMP, AMADUMPB, AMARESTART, CALLDUMP, and MAKEAMA.

When to use this document

Northern Telecom (NT) software releases are referred to as batch change supplements (BCS) and are identified by a number, for example, BCS29. This document is written for DMS-100 Family offices that have BCS35 and up.

More than one version of this document may exist. The version and issue are indicated throughout the document, for example, 01.01. The first two digits increase by one each time the document content is changed to support new BCS-related developments. For example, the first release of a document is 01.01, and the next release of the document in a subsequent BCS is 02.01. The second two digits increase by one each time a document is revised and rereleased for the same BCS.

To determine which version of this document applies to the BCS in your office, check the release information in *DMS-100 Family Guide to Northern Telecom Publications*, 297-1001-001.

How to identify the software in your office

The *Office Feature Record (D190)* identifies the current BCS level and the NT feature packages in your switch. You can list a specific feature package or patch on the MAP (maintenance and administration position) terminal by typing

```
>PATCHER;INFORM LIST  identifier
```

and pressing the Enter key.

where
identifier   is the number of the feature package or patch ID

You can identify your current BCS level and print a list of all the feature packages and patches in your switch by performing the following steps. First, direct the terminal response to the desired printer by typing

```
>SEND  printer_id
```

and pressing the Enter key.

where
printer_id is the number of the printer where you want to print the data

Then, print the desired information by typing

```
>PATCHER;INFORM LIST;LEAVE
```

and pressing the Enter key.

Finally, redirect the display back to the terminal by typing

```
>SEND PREVIOUS
```

and pressing the Enter key.

Where to find information

The chart below lists the documents that you require to understand the content of this document, or to perform the tasks it describes. These documents are also referred to in the appropriate places in the text.

More than one version of these documents may exist. To determine which version of a document applies to the BCS in your office, check the release information in *DMS-100 Family Guide to Northern Telecom Publications*, 297-1001-001.

Number	Title
297-1001-119	*Automatic Message Accounting - Northern Telecom Format*
297-1001-451	*Customer Data Schema*
297-1001-509	*Command Reference Manual*
297-1001-571	*Device Independent Recording Package (DIRP) Alarm Clearing Procedures*
297-1001-572	*Device Independent Recording Package (DIRP) Routine Maintenance Procedures*
297-1001-573	*Device Independent Recording Package (DIRP) Trouble Locating and Clearing Procedures*
297-1001-574	*Device Independent Recording Package (DIRP) Recovery Procedures*
297-1001-820	*Nonmenu Commands Reference Manual*
297-1001-821	*Menu Commands Reference Manual*
297-1001-830	*Bellcore Format Automatic Message Accounting Reference Guide*
297-2071-119	*Meridian Digital Centrex Station Message Detail Recording Reference Guide*

Number	Title
TAM-1001-000	*Listing of Technical Assistance Manuals*
TAM-1001-001	*TAS Non-Res Tool Listing Technical Assistance Manual*

What precautionary messages mean

Danger, warning, and caution messages in this document indicate potential risks. These messages and their meanings are listed in the following chart.

Message	Significance
DANGER	Possibility of personal injury
WARNING	Possibility of equipment damage
CAUTION	Possibility of service interruption or degradation

Examples of the precautionary messages follow.

DANGER
Risk of electrocution
The inverter contains high voltage lines. Do not open the front panel of the inverter unless fuses F1, F2, and F3 have been removed first. Until these fuses are removed, the high voltage lines inside the inverter are active, and you risk being electrocuted.

WARNING
Damage to backplane connector pins
Use light thumb pressure to align the card with the connectors. Next, use the levers to seat the card into the connectors. Failure to align the card first may result in bending of the backplane connector pins.

CAUTION
Loss of service
Subscriber service will be lost if you accidentally remove a card from the active unit of the peripheral module (PM). Before continuing, confirm that you are removing the card from the inactive unit of the PM.

How commands, parameters, and responses are represented

Two command conventions exist:

- command expansion - representations of commands including all parameters, variables and syntactic characteristics
- command example - representations of commands as they are entered

Command expansion conventions

A command table is used for a command expansion. This table consists of the following two sections:

- the command expansion, which contains
  - all parameters
  - all variables
  - hierarchy (the order in which elements must be entered)
  - syntax
  - truncated and abbreviated forms when allowed
  - defaults
- the parameter and variable descriptions. This section follows the command expansion and contains an alphabetical listing of all parameters and variables with a description of each.

Command elements are represented exactly as they are entered, except when Italic font is used to indicate that an element is a variable name or a certain default.

Commands

The command is represented in bold type. When commands are not case-sensitive, they are in lowercase.

The command appears to the left of all other elements (parameters and variables).

When truncated or abbreviated forms of a command are allowed, they appear directly beneath the long form of the command.

Parameters

Parameters are represented in unbolded type. When parameters are not case-sensitive, they are in lowercase.

Variables

Variables are represented in italic. Italics indicates that the variable, as represented, is not entered, but replaced with an element, a value, range, number, or item from a list.

The numbers, values, ranges, and lists are described in detail for each variable in the parameters and variables description section below the expansion.

Hierarchy

The order in which command elements are entered is represented by their order of appearance, from left to right. When several elements appear in a vertical list, only one of them may be selected for that position.

Defaults

A default parameter is underlined.

The action the system takes when an element in a vertical list is not required is called a default action, and is usually an action indicated by one of the elements that can be selected. Occasionally, the default action is something different than one indicated. These non-selectable defaults are represented by the word, “default,” in italics, to indicate that it is never entered. The default is then described in the parameters and variables section.

Related groups of elements

When an element is directly followed by another element, the second element is required when the first element is selected.

To distinguish which elements relate to which, brackets surround those elements that, as a group, pertain to other elements. Only those elements that horizontally directly precede or follow the brackets are related to the elements within the brackets. When elements are not in brackets, only those elements that directly precede or follow them are related.

The following is an example of a command expansion.

bsy command parameters and variables

Command	Parameters and variables			
bsy	link	*ps_link*	*noforce*	*wait*
	pm		force	nowait
	unit	*unit_no*		

Parameters and variables	Description
force	This parameter overrides all other commands and states in effect on the specified units. If the whole PM is to be taken out-of-service, confirmation, yes or no, is required.
link	This parameter busies one of the P-side links specified by *ps_link.*

bsy command parameters and variables (continued)

Parameters and variables	Description
noforce	This parameter indicates default condition when “force” is not entered.
nowait	This parameter enables the MAP to be used for other command entries before bsy force is confirmed. Nowait is used only with force.
pm	This parameter busies both units of the peripheral module.
ps_link	This variable specifies which of the P-side links is to be busied. Range is 0 to 3.
unit	This parameter busies one unit of the PM specified by *unit_no.*
unit_no	This variable specifies which unit of the PM is to be busied. Range is 0 to 1.
wait	This parameter indicates default condition when “nowait” is not entered.

Command examples

Command examples use the same conventions as a command expansion, except that all command elements are bold and are entered just as represented. If the variable is shown with a value, it is entered exactly like a command or parameter. If the variable name is used, it is in bold italics to indicate that it is not entered as represented. The following two examples illustrate this difference.

- This is a command example containing a variable name.

  bsy link *ps_link*

  and pressing the Enter key. ↵

- This is a command example containing a variable value.

  bsy link 2

  and pressing the Enter key. ↵

AMA tools overview

This chapter describes the AMADUMP, AMADUMPB, AMARESTART, CALLDUMP, and MAKEAMA utilities, their uses, and their restrictions and limitations. This chapter also gives general usage ideas.

Although all the tools described in this document do have the ability to influence Device Independent Recording Package (DIRP) files, DIRP is not described in this document. For a full description of DIRP, see the following:

297-1001-571	*Device Independent Recording Package (DIRP) Alarm Clearing Procedures*
297-1001-572	*Device Independent Recording Package (DIRP) Routine Maintenance Procedures*
297-1001-573	*Device Independent Recording Package (DIRP) Trouble Locating and Clearing Procedures*
297-1001-574	*Device Independent Recording Package (DIRP) Recovery Procedures*

Also, the internal buffer which holds call information prior to the information being sent to DIRP is affected by all the tools. AMADUMP can access the buffer through the calldump option. CALLDUMP works directly with the buffer. MAKEAMA creates information in the buffer. AMADUMPB empties the buffer to DIRP.

The buffer is important to the user because it holds the call information prior to the information being sent to DIRP. If a user is working in the buffer and the buffer is cleared, the information is lost from the buffer and transferred to the DIRP file. The buffer will be cleared when CALLDUMP is used. The buffer will be cleared if it becomes full. The buffer will be cleared if the fields TIMERDMP and TIMERINT in Table CRSFMT are used. AMADUMPB also clears the buffer. If one of these events occurs while the user is making calls and using CALLDUMP, the user may get unexpected results.

AMADUMP

AMADUMP provides the ability to display call information from DIRP in either a formatted or unformatted form. AMADUMP handles billing records in multiple formats, the most used of which are NT and Bellcore formats. A list of all of the files that may be displayed using AMADUMP may be found by listing table CRSFMT.

When to use AMADUMP

Use AMADUMP when the information is located in a Device Independent Recording Package (DIRP) file, or when the amount of calls generated by the switch requires that the filter option be used. Generally, AMADUMP should be used if the switch handles a large volume of calls.

AMADUMP modules and subsystems

AMADUMP is packaged in a single module and subsystem. The module must be present for AMADUMP to function.

Table 1-1 lists the subsystems and modules associated with AMADUMP. If a command or function fails to operate correctly, refer to table 1-1 to find the module or subsystem required to perform the command and then check the load to see if the subsystem is included.

Table 1-1xxx
AMADUMP modules and subsystems

Subsystem	Module, function, and comments
AMASUB	AMASUB is the AMADUMP subsystem. AMASUB contains the following AMADUMP module. AMADMPCI The complete AMADUMP module.

AMADUMP interaction

AMADUMP allows only a single user to access a file. Therefore, user interaction in AMADUMP should be minimal. There is a chance, however, for one user to affect another. If a user dumps the holding buffer (through the use of another tool, such as AMADUMPB) while a second user is using the amadump calldump command any call information - including a call that might have been desired - is not accessible with the the calldump option.

AMADUMP usage notes

The following comments are helpful when using AMADUMP.

Finding filenames

The DIRP files containing the AMA billing information can be found by accessing the DIRP level through the IOD level of the MAP display. Once at the DIRP level, a query command will display the volume name corresponding to AMA, SMDR, OM, or another selected option. Once the volume name is known, use DSKUT or DISKUT to find the file name. Note that AMADUMP only displays billable DIRP streams. To determine if a stream is billable, check table CRSFMT.

AMADUMPB

AMADUMPB dumps the current output buffer for a specified stream to the output file. The output file receiving the information is the same file that normally holds the buffer after the buffer has filled. AMADUMPB simply transfers the data prior to the buffer being filled. Once the AMADUMPB command has been issued, the data can be accessed with the AMADUMP tool.

When to use AMADUMPB

Use AMADUMPB when the user would prefer to view the call records with the AMADUMP tool rather than the CALLDUMP tool. This is useful for viewing the information over a period of time or to take advantage of the filter commands available with the AMADUMP utility. Once the AMADUMPB tool is used, the information is located in a stable environment where it can be accessed as needed.

AMADUMPB modules and subsystems

AMADUMPB is packaged in a single module and subsystem. The module must be present for AMADUMPB to function.

Table 1-2 lists the subsystems and modules associated with AMADUMPB. If a command or function fails to operate correctly, reference to table 1-2 to find the module or subsystem required to perform the command and then check the load to see if the subsystem is included.

Table 1-2xxx
AMADUMPB modules and subsystems

Subsystem	Module, function, and comments
AMASUB	AMASUB is the AMADUMPB subsystem. AMASUB contains the following AMADUMPB module. AMACI The complete AMADUMPB module.

AMADUMPB interaction

AMADUMPB does not provide an opportunity for interaction among users; however, the possibility exists for a user to affect another user. This interaction can occur when one user requests that the current buffer be dumped to the output file just prior to another user requesting a display of the buffer. Once dumped, the buffer is empty, and the display does not include any of the desired information.

The use of the AMADUMPB tool will not permanently affect the user attempting to display the buffer. It will, however, force the user to access the information through the AMADUMP utility.

AMARESTART

AMARESTART provides the ability to re-create an AMAPROC child process. List table CRSFMT for a listing of each stream.

When to use AMARESTART

Use AMARESTART when an AMA software failure prevents a particular stream from working correctly. If a user fails to use the AMARESTART tool, the system attempts to re-create the AMA stream process once every 10 minutes.

AMARESTART modules and subsystems

AMARESTART is packaged in a single module and subsystem. The module must be present for AMARESTART to function.

Table 1-3 lists the subsystems and modules associated with AMARESTART. If a command or function fails to operate correctly, refer to table 1-3 to find the module or subsystem required to perform the command and then check the load to see if the subsystem is included.

Table 1-3xxx
AMARESTART modules and subsystems

Subsystem	Module, function, and comments
AMASUB	AMASUB is the AMARESTART subsystem. AMASUB contains the following AMARESTART module. AMARST The complete AMARESTART module.

AMARESTART interaction

AMARESTART does not provide the opportunity for interaction among users. Furthermore, any incidental interaction should be considered irrelevant; since if any interaction occurs from a necessary restart, any data being collected by another user will - in all likelihood - need to be collected again anyway. The only risk of negative interaction occurs when a user is successfully collecting data for a call at the same time a restart is performed. In this case, the restart might reinitialize a process, and call data may be lost.

CALLDUMP

CALLDUMP displays the current call information from various DIRP streams as it is dumped from the buffer to the file. One stream exists for each AMAPROC process defined in table CRSFMT. Two common streams are AMA and SMDR. See *Customer Data Schema,* 297-1001-451, for more detail on streams.

CALLDUMP offers several display formats. Users may choose between two types of formatted output, and one unformatted output. Generally, the formatted output is easier to use, but on occasion the system cannot format the output and the unformatted version must be used.

CALLDUMP is safe to use. It does not affect billing records and does not interrupt DIRP recording in progress. CALLDUMP's only effect on DIRP is that it dumps the internal call record buffer to DIRP before the buffer is full. CALLDUMP's real-time impact on call-processing is minimal.

When to use CALLDUMP

CALLDUMP is useful on low-volume switches, such as in a lab, where the current buffer does not fill quickly. If the switch volume is not fairly low, CALLDUMP is less effective since the calls desired are in the current buffer, but reside in a file. AMADUMP should be used when this occurs.

CALLDUMP modules and subsystems

CALLDUMP is packaged in a single module and subsystem. The module must be present for CALLDUMP to function.

Table 1-4 lists the subsystems and modules associated with CALLDUMP. If a command or function fails to operate correctly, refer to table 1-4 to find the module or subsystem required to perform the command and then check the load to see if the subsystem is included.

Table 1-4xxx
CALLDUMP modules and subsystems

Subsystem	Module, function, and comments
AMASUB	AMASUB is the CALLDUMP subsystem. AMASUB contains the following CALLDUMP module. CALLDUMP The complete CALLDUMP module.

CALLDUMP interaction

CALLDUMP limits multiple users in two ways. First, if a user is presently displaying the buffer, no other user can initiate a display. Users must wait until the first display has completed. Second, CALLDUMP warns users when a user has invoked CALLDUMP recently. This warning does not stop a user from displaying the buffer. It only warns the user of possible interference.

CALLDUMP usage notes

CALLDUMP can be accessed through AMADUMP by typing the appropriate command. If accessed in this fashion, the AMADUMP filters can be applied to CALLDUMP.

CALLDUMP dumps the internal call record buffer to DIRP and the MAP device, and clears the buffer. This allows the next invocation of CALLDUMP to display only records generated since the previous invocation of CALLDUMP. To redisplay the records shown by the last invocation of CALLDUMP, then the calldump previous command must be entered. The calldump previous command allows the user to redisplay the same set of records as many times as desired.

MAKEAMA

MAKEAMA provides the ability to generate billing records in the internal buffer. The records are generated without making a phone call. MAKEAMA processes the records through AMAPROC with the other AMA records, but distinguishes the generated records by setting the test call flag in the study indicator.

When to use MAKEAMA

Use MAKEAMA to generate billing records for testing. The records may be accessed through CALLDUMP or AMADUMP.

MAKEAMA modules and subsystems

MAKEAMA is packaged in a single module and subsystem. The module must be present for MAKEAMA to function.

Table 1-5 lists all subsystems and modules associated with MAKEAMA. If a command or function fails to operate correctly, refer to table 1-5 to find the module or subsystem required to perform the command, and then check the load to see if the subsystem is included.

Table 1-5xxx
MAKEAMA modules and subsystems

Subsystem	Module, function, and comments
ATATSUB	ATATSUB is the MAKEAMA subsystem. ATATSUB contains the following MAKEAMA module. AMARUCI The complete MAKEAMA module.

MAKEAMA interaction

MAKEAMA does not provide for any user interaction, unless the records generated cause the internal buffer to fill. If this happens, the buffer is dumped to DIRP, and the possibility of affecting another user exists. Also, extensive use of MAKEAMA may slow other users who are searching the call buffer for a specific call.

Online help

If preceded by any parameter, all commands print a help message. The help message shows the syntax of the command, and lists all subcommands.

Interpreting output

This chapter outlines how the AMADUMP and the CALLDUMP output appears. This document does not detail fields shown in the output samples. The specific values of each field are outlined in other documents. For additional information on each of the fields in the NT format see *Automatic Message Accounting - Northern Telecom Format*, 297-1001-119. Information on the meaning of each field in the Bellcore format is available in *Bellcore Format Automatic Message Accounting Reference Guide*, 297-1001-830. Information on the Station Message Detail Recording (SMDR) fields is available in *Meridian Digital Centrex Station Message Detail Recording Reference Guide*, 297-2071-119.

AMADUMP displays

AMADUMP displays call and data information, the filters being applied to the records, and an initial screen showing the call event information for NT format records. The call record and data record information can be shown in a variety of forms through the use of the dump command, but the filter display and intial NT format display have only one form.

NT format initial display

When AMADUMP is invoked with the Northern Telecom format, a table is displayed for the user. This table outlines call event information for the call with a Yes/No system. A sample is shown in figure 2-1. For a complete description of each of the fields in the table, see *Automatic Message Accounting - Northern Telecom Format,* 297-1001-119.

Figure 2-1xxx
Sample table for NT format

```
         INFORMATION DIGITS TABLE

  DIGIT EVENT                VALUE
                        0 1 2 3 4 5 6 7
  1     SERV OBSERVED?  N Y N Y N Y N Y
        CHARGE?         N N Y Y N N Y Y
        TRAFF SAMPLED?  N N N N Y Y Y Y
  2     ANI FAIL?       N Y N Y N Y N Y
        OP DIALED?      N N Y Y N N Y Y
        OP IDENTIFIED?  N N N N Y Y Y Y
(The above table applies to NT AMA data only-NOT SMDR data)
AMADUMP >>>
```

AMADUMP dump command displays

AMADUMP output consists of three sections: a header, the call information, and a trailer. The header precedes the call information, and the trailer follows the information. The call information is split and displayed according to the block number and the option selected.

The dump command offers several display options. The user may choose from the call record display, the data record display, a hexadecimal display of the call record, and a header display. Additionally, all displays offer a choice between a shortened version and a more detailed version.

Call option display

The call option with details is the default form of output for AMADUMP. In figure 2-2, the header appears at the start, and the trailer appears at the end. Between the header and the trailer, two calls appear. Both calls reside in the first block of memory, and the start of each is signaled by an asterisk (*). The fields in each call follow a short field definition and conclude with the capital letter *C*.

Figure 2-2xxx
Sample AMADUMP call output with details option

```
>>>BC AMA FILE U920911082803AMA IS BEING PROCESSED.

>>>BLOCK NO: 1
* HEX ID:AA   STRUCTURE CODE:00500C  CALL CODE:006C  SENSOR TYPE:036C
   SENSOR ID:0000000C   REC OFFICE TYPE:036C   REC OFFICE ID:0000000C
   DATE:20917C   CLD PTY OFF-HK:0C   SERVICE FEATURE:000C  ORIG NPA:613C
   ORIG NUMBER:6211233C   OVERSEAS IND:0C   TERM NPA:00613C
   TERM NUMBER:5551212C   CONNECT TIME:1625163C  ELAPSED TIME:000000104C

* HEX ID:AA   STRUCTURE CODE:40500C  CALL CODE:006C  SENSOR TYPE:036C
   SENSOR ID:0000000C   REC OFFICE TYPE:036C   REC OFFICE ID:0000000C
   DATE:20917C   CLD PTY OFF-HK:1C   SERVICE FEATURE:000C  ORIG NPA:613C
   ORIG NUMBER:6211233C   OVERSEAS IND:0C   TERM NPA:00613C
   TERM NUMBER:5551212C   CONNECT TIME:1626300C  ELAPSED TIME:000000000C
   MODULE CODE:025C   CIRCUIT DATE:20917C   CIRCUIT TIME:1626350C
   MODULE CODE:000C

>>>END OF FILE: U920911082803AMA
```

Users may choose to not include the definition of each field in the display by using the nodetails option. If nodetails is selected, the output appears much like the sample in figure 2-3. Notice how the fields correspond to the fields in figure 2-2 with the only difference being the text definition preceding each field.

Figure 2-3xxx
Sample AMADUMP call output with nodetails option

```
>>>BC AMA FILE U920911082803AMA IS BEING PROCESSED.

>>>BLOCK NO: 1
AA 00500C 006C 036C 0000000C 036C 0000000C 20917C 0C 000C 613C 6211233C
0C 00613C 5551212C 1625163C 000000104C

AA 40500C 006C 036C 0000000C 036C 0000000C 20917C 1C 000C 613C 6211233C
0C 00613C 5551212C 1626300C 000000000C 025C 20917C 1626350C 000C

>>>END OF FILE: U920911082803AMA
```

Data option display

The AMADUMP data display applies to Northern Telecom format only and appears much like the call record display. The display begins with a header and ends with a trailer. The information appears in block form with blocks being separated by block headers. For additional information on each of the fields in the NT format see *Automatic Message Accounting - Northern Telecom Format,* 297-1001-119. Figure 2-4 shows a sample with the details option. Figure 2-5 shows a sample with the nodetails option.

Figure 2-4xxx
Sample AMADUMP data output with details option

```
>>>NT AMA FILE R930108014606SMDR IS BEING PROCESSED.

*DATA BLK HDR ID:C2C2  DAY=008  TIME: HR=01  DATA BLK COUNT: 00000
 OFC ID:000000  INTERCHANGE FMT:0  REC LEN:030

*REC CODE:     REQ SEC:00000  SPACE:

*REC CODE:K    SPACE:    GRPID:0227  SPACE: CLLI:EADNMTG1

*REC CODE:     REQ SEC:00001  SPACE:

*REC CODE:K    SPACE:    GRPID:0226  SPACE: CLLI:EADNMTG2

*REC CODE:     REQ SEC:00003  SPACE:
     .
     .
     .
*REC CODE:     REQ SEC:00020  SPACE:

*REC CODE:K    SPACE:    GRPID:0620  SPACE: CLLI:AIS

*REC CODE:     REQ SEC:00021  SPACE:

*REC CODE:K    SPACE:    GRPID:0619  SPACE: CLLI:ANA

*REC CODE:     REQ SEC:00022  SPACE:
     .
     .
     .
```

Figure 2-5xxx
Sample AMADUMP data output with nodetails option

```
NT AMA FILE R930108014606SMDR IS BEING PROCESSED.

C2C2 008 01 00000 000000 0 030
   00000
K    0227 EADNMTG1
   00001
K    0226 EADNMTG2
   00002
K    0225 EADNMTG3
   00003
     .
     .
     .
   00053
K    0541 LONDON4902T0
   00054
K    0540 LONDON4902T1
   00055
K    0539 LOOPA1
   00056
K    0538 LOOPA2
   00057
K    0537 LTU
   00058
K    0536 MONTALK
   00059
K    0535 MTRLPQ0201T0
   00060
K    0534 MTRLPQ0201T1
   00061
K    0533 MTRLPQ0201T2
   00062
K    0532 MTRLPQ0201T3
   00063
K    0531 MTL90AD
     .
     .
     .
```

Hdr (header) option display

Generally, an AMADUMP header precedes the output. Users may also print just the header by using the header option. The header informs the user of the format of the file, the type of the information contained in the file, and the name of the file. Figure 2-6 shows a sample header. In this sample, the file is formatted as Bellcore, it is an AMA file, and it is named U920911082803AMA.

Figure 2-6xxx
Sample AMADUMP hdr output

```
>>>BC AMA FILE U920911082803AMA IS BEING PROCESSED.
```

Hex option display

If the system cannot format the output, the user may still see the output in an unformatted form. Figure 2-7 shows how the output appears in binary coded decimal.

When printed in this fashion, the calls are not separated. The user is limited in field recognition by the capital letter *C*, which denotes the end of a particular field. If the system cannot format the data, however, it is likely that some of the information bits are incorrect and that the letters shown are also incorrect.

Figure 2-7xxx
Sample AMADUMP call output with hex option

```
>>>BC AMA FILE U920911082803AMA IS BEING PROCESSED.

>>>BLOCK NO: 1
025B 0000 0033 0000 AA00 500C 006C 036C 0000 000C 036C 0000 000C 2091 7C0C
000C 613C 6211 233C 0C00 613C 5551 212C 1625 163C 0000 0010 4C00 3E00 00AA
4050 0C00 6C03 6C00 0000 0C03 6C00 0000 0C20 917C 1C00 0C61 3CC6 2112 33C0
C006 13C5 5512 12C1 6263 00C0 0000 0000 C025 C209 17C1 6263 50C0 00C

>>>END OF FILE: U920911082803AMA
```

AMADUMP filter command displays

The AMADUMP filter display command displays the entries in the filter table. A sample display is shown in figure 2-8.

Figure 2-8xxx
Sample CALLDUMP output without options

```
Filter  Field                   Filter         Filter
entry   name                    value          attributes
---------------------------------------------------------
   1    STRUCTURE_CODE          00076          AND EQ
   2    CALL_CODE               003            AND EQ

Note: Filtering is currently disabled.
```

Each entry in the table includes the entry number, the name of the field being screened, the value being screened for, and the logical functions to apply to the value.

The filter entry number is shown on the left edge of the display. The number is created when the corresponding row is added to the table. There is a unique entry number for each entry in the table. The removal of an entry does not change later numbers.

The filter name displays the field to be screened. The corresponding field in each record is screened according to the filter value. The screening choice is shown under the filter attributes heading. The type of screening performed is based on the field attributes. The user may choose to display any record that has a field value equal to (EQ), not equal to (NEQ), less than or equal to (LTE), or greater than or equal to (GTE) the filter value. The choice of how to screen each record is made when the entry is added to the table.

Also shown under the field attributes heading is a second logical operation. To display a record, either all filters with an *and* attribute must be met, or at least one filter with an *or* attribute must be met.

The filter table also shows the status of the screening. The screening is either enabled or disabled. If it is enabled, the filters are applied to the records. If the screening is disabled, the filters are not applied.

CALLDUMP displays

CALLDUMP provides three forms of output: formatted, unformatted, and fully formatted. Formatted is the default. The other two options may be specified in the command string.

Unformatted (hex) display

In some cases the system cannot format or can only partially format the output into the appropriate fields. When this occurs users may display the binary coded decimal for each field. Figure 2-9 shows a sample of the hex format.

Figure 2-9xxx
Sample CALLDUMP output with hex option

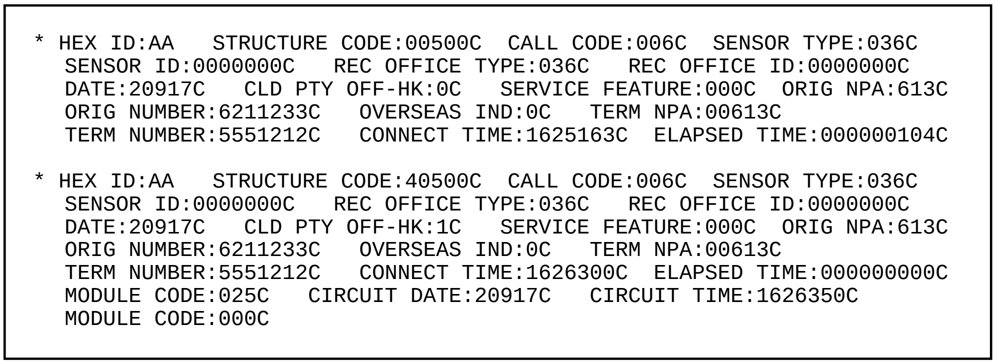
```
025B000000330000AA00500C006C036C0000000C036C0000000C20917C0C000C613C621123
3C0C00613C5551212C1625163C000000104C003E0000AA40500C006C036C0000000C036C00
00000C20917C1C000C613CC6211233C0C00613C5551212C1626300C000000000C025C20917
C1626350C000C
```

The sample shown contains the same data as figure 2-10. Only the format differs. Like the formatted version, a capital letter *C* signals the end of each field. Users should be cautious as they examine the hex output. It is possible that the system was unable to correctly format the output because of a small error that offset all the bits, or set one of the values to a *C* instead of a data value. If this happens, users may read the wrong data value.

Partially formatted (default) display

If the user does not select an output option, CALLDUMP defaults to a formatted output. A sample is shown in figure 2-10.

Figure 2-10xxx
Sample CALLDUMP output without options



```
* HEX ID:AA   STRUCTURE CODE:00500C  CALL CODE:006C  SENSOR TYPE:036C
  SENSOR ID:0000000C   REC OFFICE TYPE:036C   REC OFFICE ID:0000000C
  DATE:20917C   CLD PTY OFF-HK:0C   SERVICE FEATURE:000C  ORIG NPA:613C
  ORIG NUMBER:6211233C   OVERSEAS IND:0C   TERM NPA:00613C
  TERM NUMBER:5551212C   CONNECT TIME:1625163C  ELAPSED TIME:000000104C

* HEX ID:AA   STRUCTURE CODE:40500C  CALL CODE:006C  SENSOR TYPE:036C
  SENSOR ID:0000000C   REC OFFICE TYPE:036C   REC OFFICE ID:0000000C
  DATE:20917C   CLD PTY OFF-HK:1C   SERVICE FEATURE:000C  ORIG NPA:613C
  ORIG NUMBER:6211233C   OVERSEAS IND:0C   TERM NPA:00613C
  TERM NUMBER:5551212C   CONNECT TIME:1626300C  ELAPSED TIME:000000000C
  MODULE CODE:025C   CIRCUIT DATE:20917C   CIRCUIT TIME:1626350C
  MODULE CODE:000C
```

Although the specific fields shown in the sample are Bellcore format, certain characteristics run through all the CALLDUMP output formats. First, the start of each individual call is signalled by an asterisk (*) on the left hand side of the display. Second, each field has a short text description prior to the data. And, third, the end of each field is signalled by the capital letter *C*. This letter is not considered to be part of the data, but is used by the system to recognize the end of each field.

Fully formatted (full) display

The final output form available is the most complete. Like the formatted form, fully formatted output begins each data field with a text definition of the field. The data fields follow the definition, again with each field ending in a capital letter *C*. Additionally, a third field contains a verbal description of what the data represent. This third field appears when possible. Figure 2-11 shows a sample of the output.

Figure 2-11xxx
Sample CALLDUMP output with full option

```
*
  HEX ID:                  AA
  STRUCTURE CODE:          00500C
  CALL CODE:               006C           STATION PAID
  SENSOR TYPE:             036C           DMS 100F
  SENSOR ID:               0000000C
  REC OFFICE TYPE:         036C           DMS 100F
  REC OFFICE ID:           0000000C
  DATE:                    20917C         SEPTEMBER 17,1992
  CLD PTY OFF-HK:          0C             CLD OFF-HOOK DETECTED
  SERVICE FEATURE:         000C           OTHER
  ORIG NPA:                613C
  ORIG NUMBER:             6211233C
  OVERSEAS IND:            0C             NPA DIALED
  TERM NPA:                00613C
  TERM NUMBER:             5551212C
  CONNECT TIME:            1625163C       16:25:16.3
  ELAPSED TIME:            000000104C     000000:10.4

*
  HEX ID:                  AA
  STRUCTURE CODE:          40500C
  CALL CODE:               006C           STATION PAID
  SENSOR TYPE:             036C           DMS 100F
  SENSOR ID:               0000000C
  REC OFFICE TYPE:         036C           DMS 100F
  REC OFFICE ID:           0000000C
  DATE:                    20917C         SEPTEMBER 17,1992
  CLD PTY OFF-HK:          1C             CLD OFF-HOOK NOT DETECTED
  SERVICE FEATURE:         000C           OTHER
  ORIG NPA:                613C
  ORIG NUMBER:             6211233C
  OVERSEAS IND:            0C             NPA DIALED
  TERM NPA:                00613C
  TERM NUMBER:             5551212C
  CONNECT TIME:            1626300C       16:26:30.0
  ELAPSED TIME:            000000000C     000000:00.0
  MODULE CODE:             025C           CIRCUIT RELEASE
  CIRCUIT DATE:            20917C         SEPTEMBER 17,1992
  CIRCUIT TIME:            1626350C       16:26:35.0
  MODULE CODE:             000C           FINAL MODULE
```

AMADUMP level commands

Use the AMADUMP level of the MAP display to display or print the contents of the automatic message accounting (AMA) files produced in local or centralized AMA offices. The following formats may be used:

- block-by-block hexadecimal dump of the file contents for a specified range of blocks
- record-by-record dump of AMA call entries, data entries, or header entries in an AMA file with or without screening specified

See the appropriate section in chapter 2, “Interpreting output,” for an explanation of each of the formats.

CAUTION
Possible loss of AMA
If the file being accessed through AMADUMP is a processed file on a mounted volume, DIRP might attempt to erase the file and end up doing an emergency rotate on the volume.

The emergency rotate will result from the file being open when DIRP accesses it. DIRP will only attempt to erase the file if additional space is needed on the volume, and the file is one of the processed (P) files chosen by DIRP for erasure. DIRP selects files files for erasure according to the age of the file. The oldest processed files are erased first.

To avoid any chance of an emergency rotate, demount the volume prior to using AMADUMP on the file. If the volume cannot be demounted and must remain mounted, ensure a minimum of 2 Mbytes is available on the volume during the whole course of AMADUMP.

Accessing the AMADUMP level

To access the AMADUMP level, enter the following command from the CI level:

```
amadump format filename ↵
```

The *format* variable specifies the form in which the data are transmitted and stored. Valid formats are NT, INTL, CDR, CDRA, CDRB, CDRC, CDRCTEMP, CDRD, VCDRUCS26, and BC. The most common format is Bellcore (BC).

The *filename* variable specifies the name of the file to be worked with, or calldump. The file name may be any AMA, SMDR, or other billable file resident on the volume. If calldump is entered, the amadump command applies the CALLDUMP parameters. See the CALLDUMP CI level tool for an explanation of the calldump command.

Finding filenames

The Device Independent Recording Package (DIRP) files containing the AMA billing information can be found by accessing the DIRP level through the IOD level of the MAP display. Once at the DIRP level, a query command displays the volume name corresponding to AMA, or SMDR. Once the volume name is known, use the DSKUT utility or the DISKUT utility to find the file name. Note that AMADUMP only displays billable files. The billable files may be seen by listing table CRSFMT.

AMADUMP commands

All the commands available at the AMADUMP MAP level are described in this chapter and are arranged in alphabetical order. The page number for each command is listed in the following table.

AMADUMP commands

Command		**Page**
dump	AMADUMP	3-3
filter	AMADUMP	3-11
help	AMADUMP	3-21
quit	AMADUMP	3-23

dump

Function

Use the dump command to display call information from DIRP. The display may be in either a formatted or unformatted form.

dump command parameters and variables

Command	Parameters and variables			
dump	call data hdr hex	[[*nodetails* details]	*startblk*	*numblcks*]

Parameters and variables	Description
call	This parameter dumps header entries, control entries, and call entries in the C1C1 call record block. The block is dumped in ASCII. The data also includes associated call extension entries when they are present.
data	This parameter dumps header entries, control entries, and data entries in the C2C2 data blocks. This information is not call related; it generally represents operational measurements taken over a set period of time.
details	This parameter dumps records in detailed format as follows: ▪ provides each field in the record associated with the field name ▪ appends additional information to the end of each record indicating if the call type was direct dialing overseas and if the call was answered.
hdr	This parameter dumps the block header entries and control entries.
hex	This parameter causes an unformatted hexadecimal dump of block data and is applicable to all tape formats.
nodetails	If nothing is entered to specify the format, the system defaults to using the simple format with each field in the record, separated by a space.
numblcks	This variable specifies the number of blocks to be dumped. Valid entries are 1-32 000. If a value is not specified, the system displays all blocks.
startblk	This variable specifies the starting number of the blocks to be displayed. The count is from the beginning of the file and not the block count shown in the C1C1 header. Valid entries are 1-32 000. If no value is entered, the system defaults to the first block.

dump (continued)

Qualifications

The dump command is qualified by the following exceptions, restrictions, and limitations:

- Call entries printed using the dump call command string or dump data command string can be screened based on fields that satisfy certain data patterns or time constraints. The AMADUMP directory filter command is used for this purpose and must be specified prior to using the dump command.
- The dump data command cannot be used to obtain AMA record information with the Bellcore format. The dump data command applies only to NT format records.
- The dump command cannot be used if the calldump command was specified. The dump command can only be used with a DIRP file.
- Errors that occur while using the dump command are signaled by error messages that detail the reason for the failure.

Example

The following table provides an example of the dump command.

Example of the dump command

Example **Task, response, and explanation**

dump call details ↵

Task: Display call entry records in detailed format.

Response:

```
A BC AMA FILE IS BEING PROCESSED.
*HEX ID=AA  STRUCT CODE:00020C CALL TYPE:001C SENSOR TYPE:036C
SENSOR ID:0000000C  REC OFC TYPE:036C  REC OFC ID:0000000C
DATE:60422C  TIMING IND:00000C  STUDY IND:2300000C  ANSWER:0C
SERV OBSERVED:0C  OPER ACTION:0C SERV FEAT:000C  ORIG NPA:613C
ORIG NO:6211092C  OVERSEAS IND:1C  TERM NPA:00613C
TERM NO:6211234C  CONN TIME:0043506C  ELAPSED TIME:000000051C
WATS IND:0C  WATS BAND  IND:020C
```

Explanation: This example provides one call entry record of the detailed call record format.

Responses

The following table provides an explanation of the responses to the dump command.

Responses for the dump command

MAP output	Meaning and action
`BLOCK READ FAILED`	**Meaning:** Either an input/output (I/O) error has occurred or an incorrect block size was entered. **Action:** Reissue the command. If the command fails again, contact the next level of maintenance. Note the file name, the device, the date that the file was created, and the block that failed.
`COMMAND ABORTED.  COULD NOT OPEN FILE:` *<FILENAME>*	**Meaning:** The specified file does not exist. **Action:** Open an existing file with the necessary information or create a new file.
`COMMAND ABORTED.  INVALID DETAILS PARAMETER:` *<PARAMETER>*	**Meaning:** The parameter was incorrectly specified. **Action:** Reissue the command using the correct parameter.
`COMMAND ABORTED.  UNKNOWN DUMP FUNCTION:` *<FUNCTION>*	**Meaning:** The appropriate dump functions are hdr, hex, call, and data. (Entering the dump data command produces a message referring the user to the dump call command). **Action:** Reissue the command using the appropriate dump function.
`CORRUPT RECORD ENCOUNTERED.`	**Meaning:** One of the records being displayed is corrupt. The command was aborted. **Action:** Pinpoint the records being displayed to avoid the corrupt records, and retry the command with the necessary parameters.

-continued-

dump (continued)

Responses for the dump command (continued)

MAP output **Meaning and action**

```
CORRUPTED BDW <BDW> FOUND IN BLOCK NO: <BLOCK NUMBER> BDW SIZE COMPUTED
AS: <SIZE> BLOCK READ FAILED FAILURE OCCURRED WHILE DUMPING FILE:
<FILENAME>
```

Meaning: The block descriptor word of the record is corrupted.

Action: Reissue the command.

```
CORRUPTED RDW <RDW> FOUND IN BLOCK NO: <BLOCK NUMBER> RDW SIZE COMPUTED
AS: <SIZE> BAD RECORD ENCOUNTERED PROCEEDING TO NEXT BLOCK
```

Meaning: The record descriptor word, which is the first four bytes of a record, is corrupted.

Action: Reissue the command.

```
COULD NOT FORMAT BLOCK CONTAINING DEFERRED DATA. ERROR OCCURRED IN BLOCK
NO: <BLOCK NUMBER> PROCEEDING TO NEXT BLOCK.
```

Meaning: The specified block contains unformatted AMA data. The data could be corrupted or the tape may have come from a switch that does not have the optional deferred AMA feature.

Action: Ensure that the switch software can handle the information on the tape. Reissue the command. If the command fails again, contact the next level of maintenance.

```
END OF RECORD WAS ENCOUNTERED WITHOUT DETECTING MODULE CODE ZERO. RECORD
SIZE (NOT INCLUDING RDW) IS: <SIZE> IN BLOCK NO: <BLOCK NUMBER RECORD
CONTENTS> BAD RECORD ENCOUNTERED PROCEEDING TO NEXT RECORD
```

Meaning: Module code 000 is the final module in any set of modules in an AMA record. If module code 000 is missing, this error message occurs.

Action: Ensure that module code 000 is the final module in the AMA record.

```
ERROR ENCOUNTERED WHILE CLOSING FILE: <FILENAME>
```

Meaning: An internal software or hardware error of unknown origin has occurred.

Action: Contact the next level of maintenance.

-continued-

dump (continued)

Responses for the dump command (continued)

MAP output	Meaning and action
FAILURE OCCURRED WHILE DUMPING FILE: *<FILENAME>*	**Meaning:** This message accompanies other error messages explaining the failure. **Action:** Refer to accompanying error messages to determine fault causes and corrective actions.
NIL ASPECT INVOKED FOR BLOCK READ	**Meaning:** An internal software or hardware error of unknown origin has occurred. This message could appear not only for block read, but also for record read, filter, and dump. **Action:** Contact the next level of maintenance.
NO RECORDS MATCHING FILTER DATA WERE FOUND	**Meaning:** A dump of call records was requested, but based on previously requested filter values, no matching call records were found that matched. **Action:** None
REMINDER WARNING: FILTER FUNCTION IS ENABLED.	**Meaning:** The filter function is active. Only specific AMA records and associated field values are displayed. (The filter function restricts the display of AMA records and fields in those records to those associated with entered values.) **Action:** None
START BLOCK WAS NOT FOUND	**Meaning:** The start block number exceeds the number of blocks on the tape. **Action:** Reissue the command with a valid start block number.

-continued-

dump (continued)

Responses for the dump command (continued)

MAP output **Meaning and action**

```
SOFTWARE ERROR. RECORD IN BLOCK <BLOCK NUMBER> EXCEEDS MAXIMUM SIZE OF A
FILTER TABLE ENTRY <SIZE> RECORD CONTENTS: <RECORD CONTENTS> FILTER
PROCESSING IS UNAFFECTED AND CONTINUES NORMALLY
```

Meaning: The specified record is longer than the maximum size of a filter table entry.

Action: Reduce the size of the record.

```
THE DUMP DATA COMMAND IS NOT APPLICABLE TO BELLCORE AMA FORMAT.  ALL
BELLCORE AMA RECORDS ARE DISPLAYED WITH THE DUMP CALL SUBCOMMAND.
```

Meaning: The dump data command cannot be used to obtain AMA record information with the Bellcore format.

Action: Use the dump call command to obtain AMA record information.

```
TRUNCATED RECORD ENCOUNTERED RECORD SIZE (NOT INCLUDING RDW) IS: <SIZE> IN
BLOCK NO: <BLOCK NUMBER RECORD CONTENTS> BAD RECORD ENCOUNTERED PROCEEDING
TO NEXT RECORD
```

Meaning: One of the records in the file was truncated.

Action: None

```
UNSUCCESSFUL ATTEMPT TO READ BLOCK NO: <BLOCK NUMBER> FILESYS RETURN CODE:
<RETURN CODE> BLOCK READ FAILED FAILURE OCCURRED WHILE DUMPING FILE:
<FILENAME>
```

Meaning: Either an I/O error has occurred or an incorrect block size was entered.

Action: Reissue the command. If the command fails again, contact the next level of maintenance. Note the file name, device, the date that the file was created, and the block that failed.

```
UNRECOGNIZED MODULE CODE <MODULE CODE> ENCOUNTERED IN BLOCK NO: <BLOCK
NUMBER RECORD CONTENTS> BAD RECORD ENCOUNTERED PROCEEDING TO NEXT RECORD
```

Meaning: Specific module code numbers can be appended to AMA records. This error message indicates that a module code was entered that either is nonexistent or is not recognized by the software in this feature package.

Action: Reissue a correct module code.

-continued-

dump (end)

Responses for the dump command (continued)

MAP output **Meaning and action**

UNRECOGNIZED STRUCTURE CODE *<SC>* IN BLOCK NO: *<BLOCK NUMBER RECORD CONTENTS>* BAD RECORD ENCOUNTERED PROCEEDING TO NEXT RECORD

Meaning: Specific structure codes exist for AMA records. This error message indicates that a structure code was entered that either is nonexistent or is not recognized by the software in this feature package.

Action: Reissue a correct structure code.

***** WARNING ***** THIS FILE CONTAINS UNFORMATTED DATA WHICH CANNOT BE READ BY DOWNSTREAM RAO PROCESSORS

Meaning: The file contains unformatted AMA data. The data could be corrupted or may have come from a switch that does not have the optional deferred AMA feature.

Action: Ensure that the switch software can handle the information on the tape. Reissue the command. If the command fails again, contact the next level of maintenance.

-end-

filter

Function

Use the filter command to screen call billing records. Calls may be screened by specific fields and displayed with the dump command.

filter command parameters and variables

Command	**Parameters and variables**				
filter	add delete disable display enable	*field* *entry*	*value*	*logical*	*range*

Parameters and variables	**Description**
add	This parameter adds a filter to the filter table. Once the filter table has one or more filters, it may be enabled and the filters will be applied to the records.
delete	This parameter removes the specified filter from the filter table.
disable	This parameter disables filter screening. It has no effect on the filter table.
display	This parameter displays the contents of the filter table. If the optional value field is appended to the display command, all the possible field names are shown.
enable	This parameter enables record screening. Screening is based on the contents of the filter table.
entry	This variable specifies the entry to remove from the filter table. It correlates to the number shown on the left edge of the filter table. It may range from 1-50. To remove all of the entries, enter 0.
field	This variable specifies the Bellcore field name by which to filter. The field name should be entered as a character string in the exact form shown as when the fields are displayed using the filter display fields command.
logical	This variable specifies the logical operation to use with multiple filters. It may be either *and* or *or*. The default value is and. A record will be displayed if it passes all the filters with an *and* value, or if at least one filter passes with an *or* value.
range	This variable specifies the numerical range by which to filter the value. It may be eq, neq, lte, or gte. The default value is eq.
value	This variable specifies the screening values. When specifying a screening value, valid entries are 0-9, A-F , and *. The asterisk (*) acts as a wildcard.

filter (continued)

Qualifications

The filter command is qualified by the following exceptions, restrictions, and limitations:

- Adding an entry to the filter table does not enable the filter command. To enable the screening process, the user must first add an entry to the filter table, and then enable the filter.
- The filter command can define screening for call records when using the dump call command string, dump data command string, or calldump option.

Examples

The following table provides examples of this command.

Examples of the filter command

Example	Task, response, and explanation
filter display ↵	

Task: Display the contents of the filter table.

Response:

```
Filter  Field             Filter     Filter
entry   name              value      attributes
-------------------------------------------------
   1    STRUCTURE_CODE    00076      AND EQ
   2    CALL_CODE         003        AND EQ

Note: Filtering is currently enabled.
```

Explanation: The contents of the filter table are shown. The filter table contains two entries. The first entry screens for a structure_code of 00076, while the second entry screens for a call_code of 003. From the attributes is can be seen that both conditions (and) must be met for the call to be displayed. For a more detailed explanation of the filter table, see the appropriate section in chapter 2.

-continued-

Examples of the filter command (continued)

Example **Task, response, and explanation**

filter add structure_code 00625 ↵
where

structure_code	specifies the field to apply the filter to
00625	specifies the particular structure code value

Task: Add an entry to the filter table.

Response:

```
>>>Filter successfully added as filter entry 1
>>>NOTE: Filter function is currently disabled.
```

Explanation: This command adds structure code 00625 to the filter table. When the process is complete, the system displays a confirmation message indicating that the function is complete. The status is also shown.

-end-

Responses

The following tables explain the responses to the filter command. The command responses are grouped as follows:

- all formats
- filter add command string (Bellcore format)
- filter delete command string
- filter display command string (Bellcore format)
- filter enable command string and filter disable command string (Bellcore format)

filter (continued)

The following table explains the responses to any filter command string for all formats.

Responses for the filter command (all formats)

MAP output	Meaning and action
`COMMAND ABORTED. ERROR ENCOUNTERED WHILE READING FILTER FUNCTION PARAMETER.`	
	Meaning: Abort was entered in response to the prompt for a parameter. **Action:** Enter a valid parameter.
`COMMAND ABORTED. RETURN CODE ERROR:` *<RETURN CODE>*	
	Meaning: An internal software or hardware error of unknown origin occurred. **Action:** Contact the next level of maintenance.
`COMMAND ABORTED. UNKNOWN FILTER FUNCTION:` *<FUNCTION>*	
	Meaning: An invalid filter function was entered. Valid filter functions are add, delete, display, enable, and disable. **Action:** Reissue the command using a known function.

filter (continued)

The following table explains the responses to the filter add command string (Bellcore format).

Responses for the filter add command (Bellcore format)

MAP output	Meaning and action
`ADDING..... STRUCTURE CODE <SC>`	**Meaning:** The specified structure code is being added to the filter table. **Action:** None

`BAD CHARACTERS <CHARACTERS>.`
`ENTER AGAIN:`
	Meaning: Incorrect characters were entered with the filter command.
	Action: Reissue the command.

`COMMAND REJECTED. INVALID PARAMETER: <PARAMETER>`	**Meaning:** An invalid parameter was entered with the filter command. **Action:** Enter a valid parameter.
`COMMAND REJECTED.  INVALID STRUCTURE CODE: <SC>`	**Meaning:** An invalid structure code was entered. **Action:** Enter a valid structure code.
`COMMAND REJECTED STRUCTURE CODE <SC> IS ALREADY IN ENTRY <ENTRY NUMBER> OF THE FILTER TABLE.`	**Meaning:** An attempt was made to add a structure code using the filter command, but the code for the entry number is already present in the filter table. **Action:** Reissue the command using a structure code that is not known to the filter table.
`COMMAND REJECTED.  THE FILTER TABLE IS FULL. A MAXIMUM OF 10 ENTRIES ARE ALLOWED.`	**Meaning:** An attempt was made to add more than ten entries using the filter add command string. **Action:** Reissue the filter add command string with no more than ten entries.

-continued-

filter (continued)

Responses for the filter add command (Bellcore format) (continued)

MAP output **Meaning and action**

```
FILTER ADDITION ABORTED.  THE MAXIMUM LENGTH OF A FILTER ENTRY (304) HAS
BEEN EXCEEDED.
```

Meaning: A filter addition was aborted because too many fields were added to an entry.

Action: Enter the filter add command string again, ensuring that the entries do not exceed the specified maximum length.

```
INVALID MODULE CODE <MC>.  ENTER AGAIN:
```

Meaning: An invalid module code was entered.

Action: Enter a valid module code.

```
SIZE OF DATA ENTERED <SIZE> EXCEEDS SIZE OF FIELD <SIZE> ENTER AGAIN:
```

Meaning: Information entered using the filter add command string exceeded the size of a particular field.

Action: Reissue data that does not exceed field size.

```
STRUCTURE CODE <SC> HAS BEEN ADDED AS ENTRY <ENTRY NUMBER> FILTER
ADDITION COMPLETE.
```

Meaning: The specified structure code has been entered using the filter add command, and the structure code has been added to the filter table

Action: None

```
THE FILTER FUNCTION IS CURRENTLY: ON COMMAND REJECTED.  THE FILTER
FUNCTION MUST BE DISABLED BEFORE ENTRIES CAN BE DELETED.
```

Meaning: To delete entries from the file, the filter function must be disabled.

Action: Disable the filter function by entering the filter disable command string.

-end-

filter (continued)

The following table explains the responses to the filter delete command string.

Responses for the filter delete command

MAP output	Meaning and action
COMMAND ACCEPTED. ALL ENTRIES IN THE FILTER TABLE HAVE BEEN REMOVED or COMMAND ACCEPTED. STRUCTURE CODE *<SC>* HAS BEEN REMOVED FROM ENTRY *<ENTRY NUMBER>* IN THE FILTER TABLE.	**Meaning:** The filter delete command string executed successfully. **Action:** None
COMMAND ACCEPTED. STRUCTURE CODE *<SC>* HAS BEEN REMOVED FROM ENTRY *<ENTRY NUMBER>* IN THE FILTER TABLE.	**Meaning:** The filter delete command executed successfully. **Action:** None
COMMAND REJECTED. STRUCTURE CODE *<SC>* WAS NOT FOUND IN THE FILTER TABLE.	**Meaning:** An attempt was made to delete an entry from the filter table, but the table contained no entries. **Action:** None
COMMAND REJECTED. THE FILTER TABLE IS EMPTY.	**Meaning:** An invalid parameter was entered with the filter command. **Action:** Reissue the command with a valid parameter.
THERE ARE NO MORE ENTRIES IN THE FILTER TABLE.	**Meaning:** Once enabled, the filter function remains enabled until the filter disable command string is entered. **Action:** None

filter (continued)

The following table explains the responses to the filter display command string (Bellcore format).

Responses for the filter display command (Bellcore format only)

MAP output	Meaning and action
`COMMAND REJECTED.  INVALID STRUCTURE CODE:` *<SC>* `END OF FILTER DISPLAY OR FILTER ENTRY:` *<ENTRY NUMBER>* `UNRECOGNIZED STRUCTURE CODE` *<SC>* `ENCOUNTERED` *<RECORD CONTENTS>* `OR FILTER ENTRY:` *<ENTRY NUMBER>* `*EMPTY OR FILTER ENTRY:` *<ENTRY NUMBER ENTRY CONTENTS>* `END OF FILTER DISPLAY`	**Meaning:** The user entered an invalid structure code. **Action:** Reissue the command with a valid structure code.
`THE FILTER FUNCTION IS CURRENTLY: OFF`	**Meaning:** The filter function is disabled. **Action:** None
`THE FILTER FUNCTION IS CURRENTLY: ON`	**Meaning:** The user activated the filter function by using the filter enable command string. Now when call records are displayed, only those call records and fields specified in the filter table are shown. **Action:** None

filter (end)

The following table explains the responses to the filter enable command string and filter disable command string (Bellcore format).

Responses for the filter command

MAP output	Meaning and action

`COMMAND ACCEPTED.  THE FILTER FUNCTION IS NOW: ON`
`DUMP COMMANDS WILL SCREEN ON THE BASIS OF DATA IN THE FILTER TABLE.`
	Meaning: The filter function has been enabled with the filter enable command string.
	Action: None

`COMMAND REJECTED.  THERE ARE NO ENTRIES IN THE FILTER TABLE.`	**Meaning:** An attempt was made to delete an entry from the filter table, but the table contains no entries. **Action:** None
`THIS COMMAND WILL NOT DO ANYTHING.  THE FILTER FUNCTION IS ALREADY ENABLED.`	**Meaning:** The filter command string was entered more than once. **Action:** None

help

Function

Use the help command to receive online documentation for the AMADUMP directory.

help command parameters and variables

Command	Parameters and variables
help	*command_name*

Parameters and variables	Description
command_name	This optional variable specifies the command name to display online documentation for. If nothing is entered, the system defaults to displaying online documentation for this directory.

Qualifications

None

Example

The following table provides an example of the help command.

Example of the help command

Example Task, response, and explanation

help quit ↵
where

quit specifies a valid command for the AMADUMP directory

Task: Access online documentation.

Response:

```
Parameter is: <nlevels | incrname | ALL >
```

Explanation: This example typifies a response for the help command string.

help (end)

Response

The following table provides an explanation of the response to the help command.

Response for the help command	
MAP output	**Meaning and action**
`MODULE NOT LOADED OR NEEDS OTHER CI INCREMENT TO BE BUILT.`	
	Meaning: The directory you are trying to access is not loaded or must be accessed through another directory. **Action:** None

quit

Function

Use the quit command to exit from the AMADUMP environment, or from other environments.

quit command parameters and variables

Command	Parameters and variables
quit	[*n levels* *name* all]

Parameters and variables	Description
all	This parameter specifies that the system should exit from all tools and displays and return to the CI level.
n_levels	This variable specifies the number of levels to exit. The default value is 1.
name	This variable allows the user to exit from a specific environment yet remain in the other environments.

Qualifications

None

Examples

The following table provides examples of the quit command.

Examples of the quit command

Example	Task, response, and explanation
quit ↵	**Task:** Exit from the present environment. **Response:** `CI:` **Explanation:** The user exited the AMADUMP environment.

-continued-

quit (continued)

Examples of the quit command (continued)

Example	Task, response, and explanation
quit dskut ↵ *where* dskut is the name of the environment to be exited	
	Task: Exit from the DSKUT environment, but remain in the AMADUMP environment. **Response:** `AMADUMP>>>` `>` **Explanation:** The dskut environment was exited successfully. The user remains in the AMADUMP environment.

-end-

Responses

The following table provides explanations of the responses to the quit command.

Responses for the quit command

MAP output	Meaning and action
`CI:`	
	Meaning: The user returned to the CI level. **Action:** None
`QUIT -- Increment not found`	
	Meaning: The environment requested could not be exited. The system did not recognize the name of the environment requested. **Action:** Check the name of the environment to be exited. If the name entered is incorrect, retry the command. If the name is correct, check to see if the environment is active, or if it was previously exited.

-continued-

quit (continued)

Responses for the quit command (continued)

MAP output	Meaning and action
`QUIT -- Unable to quit requested number of levels`	
	Meaning: The number of levels requested was greater than the number of levels that could be exited. **Action:** Perform a quit all command, or retry the command with a smaller number of levels.

-end-

AMADUMPB CI level tool

Accessing the AMADUMPB CI level tool

The AMADUMPB CI level tool operates from the CI level. To use the tool, enter the appropriate command from the CI level.

Function

Use the AMADUMPB CI level tool to dump the current buffer, for the specified stream, to an output file. The output file used is the same output file as is normally used when the buffer is filled and dumped.

amadumpb command parameters and variables

Command	Parameters and variables
amadumpb	*stream*

Parameters and variables	Description
stream	This variable defines the stream to be dumped to an output file. One stream exists for each AMAPROC process defined in table CRSFMT. Two common streams are AMA and SMDR. If the stream is not specified, the stream defaults to the AMA stream. See *Customer Data Schema,* 297-1001-451, for more detail on streams.

Qualifications

The AMADUMPB command responds to the user by showing the name of the stream to be dumped and requires a confirmation of the dump.

Examples

The following table provides an example of the amadumpb command.

Examples of the amadumpb command

Example	Task, response, and explanation	
amadumpb ↵		
	Task:	Dump the current ama buffer to an output file.

	Response:	`AMA BUFFER WILL BE SENT TO OUTPUT FILE FOR STREAM: AMA`
		`Please confirm (“YES”,“Y”,“NO”,“N”):`

	Explanation:	The system responded by showing the stream to be dumped and requires confirmation in order to complete the command.

Responses

The following table provides explanations of the responses to the amadumpb command.

Responses for the amadumpb command

MAP output	Meaning and action	
`COMMAND ACCEPTED.  BUFFER HAS BEEN SENT TO OUTPUT FILE.`		
	Meaning:	The user confirmed the action, and the buffer was sent to the output file.
	Action:	None
`COMMAND REJECTED.  SPECIFIED STREAM IS INVALID:` *<stream>*		
	Meaning:	The stream specified in the command was not recognized by the system.
	Action:	Retry the command with a valid stream, or retry the command with the default stream.
`CURRENT BUFFER IS EMPTY.  DUMP REQUEST NOT NEEDED.`		
	Meaning:	The buffer requested in the command is empty. Since information could not be sent to the output file, the system did not complete the command.
	Action:	None

AMARESTART CI level tool

Accessing the AMARESTART CI level tool

The AMARESTART CI level tool operates from the CI level. To use the tool, enter the appropriate command from the CI level.

Function

Use the AMARESTART CI level tool to set a specific recording buffer or all call recording buffers to their initial state.

amarestart command parameters and variables

Command	Parameters and variables
amarestart	*stream*

Parameters and variables	Description
stream	This variable defines the stream to be reset. One stream exists for each AMAPROC process defined in table CRSFMT. Two common streams are AMA and SMDR. If the stream is not specified, all the recording buffers are reset. See *Customer Data Schema,* 297-1001-451, for more detail on streams.

Qualifications

None

Examples

The following table provides an example of the amarestart command.

Examples of the amarestart command

Example	Task, response, and explanation	
amarestart ama ↵		
	Task:	Restart the ama stream.

	Response:	`THIS COMMAND WILL CAUSE DEAD RECORDING PROCESSES FOR THE`
		`SPECIFIED STREAM TO BE RESTARTED.  DO YOU WISH TO PROCEED?`
		`Please Confirm (“YES” or “NO”):`
		`>yes`
		`AMAPROC HAS BEEN RESTARTED FOR STREAM: AMA`

	Explanation:	The ama stream was successfully restarted.

Responses

The following table provides explanations of the responses to the amarestart command.

Responses for the amarestart command

MAP output	Meaning and action	

`AMA DADDY PROCESS (AMAEI) DID NOT RESPOND FOR STREAM:` *`<stream>`*
`THIS IS A POTENTIAL EMERGENCY CONDITION.`
`ENTER <QUERY PROCESS AMAEI> AND INVESTIGATE.`		

	Meaning:	A severe software error has occurred.
	Action:	Contact the next level of maintenance immediately.
`AMAPROC HAS BEEN RESTARTED FOR STREAM:` *`<stream>`*		
	Meaning:	The stream restarted successfully.
	Action:	None
`AMAPROC PROCESS IS ALREADY RUNNING FOR STREAM:` *`<stream>`*		
	Meaning:	The stream is healthy. No action needs to be taken.
	Action:	None

-continued-

amarestart (continued)

Responses for the amarestart command (continued)

MAP output **Meaning and action**

```
COMMAND IGNORED.  NO ACTIVE STREAMS EXIST.
```

Meaning: Table CRSFMT has no active streams defined.

Action: None

```
COMMAND REJECTED.  SPECIFIED STREAM IS INVALID: <stream>
```

Meaning: The stream specified in the command was not recognized by the system.

Action: Retry the command with a valid stream, or retry the command with the default stream.

```
COMMAND TERMINATED DUE TO REQUEST
```

Meaning: The amarestart command was aborted during the confirmation sequence.

Action: None

```
THIS COMMAND WILL CAUSE DEAD RECORDING PROCESSES FOR THE
SPECIFIED STREAM TO BE RESTARTED.  DO YOU WISH TO PROCEED?
Please Confirm (“YES” or “NO”):
```

Meaning: The amarestart command requires a confirmation before it restarts a stream.

Action: To confirm the restart enter *YES*. To abort the command enter *NO*.

```
UNEXPECTED RESPONSE FROM AMA DADDY (AMAEI) FOR STREAM: <stream>
THIS IS A POTENTIAL EMERGENCY CONDITION.
ENTER <QUERY PROCESS AMAEI> AND INVESTIGATE.
```

Meaning: A severe software error has occurred.

Action: Contact the next level of maintenance immediately.

-continued-

amarestart (end)

Responses for the amarestart command (continued)

MAP output	Meaning and action
UNKNOWN AMAPROC PROCESS STATE FOR STREAM: *<stream>* ENTER <QUERY PROCESS AMAEI> AND INVESTIGATE.	
	Meaning: A severe software error has occurred. **Action:** Contact the next level of maintenance immediately.

-end-

CALLDUMP CI level tool

Accessing the CALLDUMP CI level tool

The CALLDUMP CI level tool operates from the CI level. To use the tool, enter the appropriate command from the CI level.

Function

Use the CALLDUMP CI level tool to display billing records held in the temporary buffer and to transfer the temporary buffer to a DIRP output file.

calldump command parameters and variables

Command	Parameters and variables	
calldump	*stream* previous	*format*

Parameters and variables	Description
format	This variable specifies the format for the output display. Options include hex and full. If left blank, the format defaults to the partially formatted version.
previous	This parameter displays the data previously displayed another time. The user may change the format.
stream	This variable defines the stream to be displayed. One stream exists for each AMAPROC process defined in table CRSFMT. Two common streams are AMA and SMDR. If the stream is not specified, the stream will default to the AMA stream. See *Customer Data Schema,* 297-1001-451, for more detail on streams.

Qualifications

CALLDUMP affects the Device Independent Recording Package (DIRP) by dumping the current call buffer to DIRP when the calldump command is executed. See the AMADUMPB CI level tool for more information on the effect of dumping the buffer prior to it being full. To reaccess the information, users must use the calldump previous command or the

calldump (continued)

AMADUMP tool. CALLDUMP has little impact on call processing real time.

To access CALLDUMP through AMADUMP enter *calldump* instead of a file name when the AMADUMP access command is entered. Once AMADUMP has been entered, filters can be applied to the CALLDUMP command by using the AMADUMP commands.

Users should coordinate their activity when using CALLDUMP. Each time the calldump command is executed, the current buffer is dumped to DIRP with the exception of the calldump previous command. Dumping the buffer erases any information contained in the current buffer, and could cause interference among users.

Examples

The following table provides an example of the calldump command.

calldump (continued)

Examples of the calldump command

Example **Task, response, and explanation**

calldump smdr full ↵
where

smdr is the stream
full specifies that the output should be fully formatted

Task: Display the records in the smdr buffer in a fully formatted form.

Response:

```
*
  HEX ID:                AA
  STRUCTURE CODE:        00500C
  CALL CODE:             006C          STATION PAID
  SENSOR TYPE:           036C          DMS 100F
  SENSOR ID:             0000000C
  REC OFFICE TYPE:       036C          DMS 100F
  REC OFFICE ID:         0000000C
  DATE:                  20917C        SEPTEMBER 17,1992
  CLD PTY OFF-HK:        0C            CLD OFF-HOOK DETECTED
  SERVICE FEATURE:       000C          OTHER
  ORIG NPA:              613C
  ORIG NUMBER:           6211233C
  OVERSEAS IND:          0C            NPA DIALED
  TERM NPA:              00613C
  TERM NUMBER:           5551212C
  CONNECT TIME:          1625163C      16:25:16.3
  ELAPSED TIME:          000000104C    000000:10.4
```

Explanation: One billing record is displayed. For a more detailed explanation of the display, see chapter 2.

Responses

The following table provides explanations of the responses to the calldump command.

Responses for the calldump command

MAP output **Meaning and action**

```
Bad parameter:
Specified stream is invalid: stream
```

Meaning: The stream specified was not recognized by the system.

Action: Check the command string entered. Retry the command with the correct stream.

calldump (continued)

Responses for the calldump command (continued)

MAP output	Meaning and action
Data error: Corrupt record encountered. (1) CALLDUMP is terminated.	**Meaning:** The record contents may be in the wrong format or otherwise corrupted. **Action:** Check the format. If the format is wrong, retry the command string with the proper parameters. If the formats match, attempt to pinpoint the corrupt records and retry the command string with parameters excluding the corrupt record.
Data error: Data in received block is in an unrecognized format. (2)	**Meaning:** The current buffer is not in Bellcore format. **Action:** None
EITHER incorrect optional parameter(s) OR too many parameter(s).	**Meaning:** One or more of the variables or parameters was entered incorrectly or in the wrong order. **Action:** Check the command string entered. Retry the command with the proper variables and parameters.

>>> THE “DUMP” COMMAND IS NOT ALLOWED WHILE IN THE
>>> “CALLDUMP” MODE. USE THE “CALLDUMP” COMMAND
>>> INSTEAD.
	Meaning: The dump command is not recognized in conjunction with calldump.
	Action: None

There are no records to display.	**Meaning:** The buffer is empty. **Action:** None

-continued-

calldump (end)

Responses for the calldump command (continued)

MAP output **Meaning and action**

```
User <user> is currently using CALLDUMP.
Please try again.
```

Meaning: Another user is presently executing the calldump command. Only one user may display the buffer at a time.

Action: Try the command again in a short time. Use the calldump previous command to view the buffer prior to the other user’s execution of the calldump command.

```
** WARNING ** : User user has invoked the CALLDUMP
   command within the last 5 minutes.  Multiple users
   should coordinate their use of this tool.
   You can also use CALLDUMP PREVIOUS to look at the
   other user’s CALLDUMPs without interfering.

Do you wish to continue?

Please confirm (“YES”, “Y”, “NO”, or “N”):
```

Meaning: Another user has used the CALLDUMP tool recently. Use of this tool by a different user can erase any data collected. Users should use care when this message appears.

Action: To continue the execution of the command, which may interfere with the other user, enter *Yes*. To abort the execution of the command, enter *No*.

-end-

MAKEAMA CI level tool

Accessing the MAKEAMA CI level tool

The MAKEAMA CI level tool operates from the CI level. To use the tool, enter the appropriate command from the CI level.

Function

Use the makeama command to generate sample Bellcore format billing records.

makeama command parameters and variables

Command	Parameters and variables
makeama	*call_code* *number* [sc *structure_code*]

Parameters and variables	Description
call_code	This variable specifies the call code for the billing record. The call code is a three digit number, from 001-999. The default call code is 006.
number	This variable specifies the number of billing records to generate. Users may select from 1-100 records. The default value is 1.
sc	This parameter specifies that a structure code is included in the record.
structure_code	This variable specifies the value of the structure code to be included in the record. The structure code is a five digit number, from 00001-99999.

Qualifications

If a structure code is used, the structure code must be valid for the call code specified. For a listing of the structure codes valid with each call code, see *Bellcore Format Auatomatic Message Accounting Reference Guide,* 297-1001-830.

makeama (continued)

Examples

The following table provides an examples of the makeama command.

Examples of the makeama command

Example **Task, response, and explanation**

makeama 003 2 ↵
where

003 is the call code
2 is the number of records to generate

Task: Generate two billing records with a call code of 003.

Response:

```
THIS COMMAND WILL CAUSE 2 BILLING RECORD(S) WITH
CALL CODE   3 TO BE GENERATED.  DO YOU WISH TO
PROCEED?
Please confirm (“YES” or “NO”):
>yes
COMMAND COMPLETED SUCCESSFULLY.
2 BILLING RECORD(S) GENERATED.
```

Explanation: After the confirmation, the command executed successfully.

makeama 119 5 sc 00653 ↵
where

119 is the call code
5 is the number of records to generate
sc is the structure code identifier
00653 is the structure code value

Task: Generate one billing record with a call code of 119 and a structure code of 00653.

Response:

```
THIS COMMAND WILL CAUSE 5 BILLING RECORD(S) WITH
CALL CODE 119 TO BE GENERATED.  DO YOU WISH TO
PROCEED?
Please confirm (“YES” or “NO”):
>yes
COMMAND COMPLETED SUCCESSFULLY.
5 BILLING RECORD(S) GENERATED.
```

Explanation: After the confirmation, the command executed successfully.

Responses

The following table provides explanations of the responses to the makeama command.

Responses for the makeama command

MAP output	Meaning and action
Call Code *<xxx>* cannot be provided for Structure *<yyyyy>*.	**Meaning:** The combination of call code and structure code requested is not supported. **Action:** None
COMMAND COMPLETED SUCCESSFULLY. *x* BILLING RECORD(S) GENERATED.	**Meaning:** The command executed correctly, creating one or more billing records. **Action:** None
COMMAND TERMINATED DUE TO REQUEST	**Meaning:** The command did not execute. The user stopped the execution of the command when asked to confirm. **Action:** None
EITHER incorrect optional parameter(s) OR too many parameters. COMMAND ABORTED. OPTIONAL PARAMETER ERROR.	**Meaning:** Check the command string entered. One or more of the parameters or variables were wrong. **Action:** Retry the command with the proper parameters and variables.
INVALID STRUCTURE CODE SPECIFIED.	**Meaning:** The structure code entered was incorrect. **Action:** Retry the command with a valid structure code.

-continued-

makeama (end)

Responses for the makeama command (continued)

MAP output **Meaning and action**

```
Out of Range: <CALL CODE> {1 TO 999}
Enter: <CALL CODE> (|<QUANTITY>|) (|<STRUCTURE CODE>|)
```

Meaning: The call code entered was out of range.

Action: Enter a call code that is within the range specifications. The number of records to be created and a corresponding structure code may also be entered.

```
STRUCTURE CODE GENERATION IS NOT SUPPORTED
FOR GENERIC CALL CODES.
```

Meaning: The combination of call code and structure code specified is not valid.

Action: Retry the command with a valid call code and structure code combination.

```
THIS COMMAND WILL CAUSE x BILLING RECORD(S) WITH CALL
CODE xxx TO BE GENERATED.  DO YOU WISH TO PROCEED?
Please confirm (“YES” or “NO”):
```

Meaning: Prior to command execution, the system needs a confirmation.

Action: To complete the execution of the command, enter *YES*. To abort the command, enter *NO*.

```
Wrong type: <STRUCT_CODE> {1 to 9999}
Enter: <STRUCT_CODE>
```

Meaning: The structure code entered was incorrect.

Action: Enter a valid structure code.

-end-

List of terms

AMA

automatic message accounting

automatic message accounting (AMA)

An automatic recording system that documents all the necessary billing data of subscriber-dialed long distance calls.

BAF

Bellcore AMA format

batch change supplement (BCS)

A DMS-100 Family software release.

Bellcore AMA format (BAF)

The standard format for automatic message accounting (AMA) data used by Bell operating companies. The format consists of a structure code that identifies the format of the data fields in the call record, a call code that identifies the type of call recorded in the call record, other data fields that define the attributes of the call, and if needed, one or more module codes that identify the format of any additional data appended to the call record. *See also* expanded Bellcore AMA format.

BCS

batch change supplement

Bell-Northern Research (BNR)

Part of the tricorporate structure consisting of Bell Canada, Northern Telecom, and Bell-Northern Research.

BNR

Bell-Northern Research

call

In a DMS switch, any demand to set up a connection through the switch. Also used as a unit of telephone traffic. Also known as a cue.

call code

A numerical value used to distinguish types of calls.

call duration

The interval of time between the moment a connection is established between the calling and called stations and the moment the calling station gives the clearing signal (or the moment that the connection is taken down by the operator).

call processing (CP)

The software that handles the processes involved in setting up connections through the DMS-100 Family network between calling and called parties.

CI

command interpreter

command

- A control signal.
- In user interface language, the specification of an expected action or function by the system.

command interpreter (CI) level

The initial MAP level where commands are entered.

CRSFMT

The table used to hold billable call streams.

cue

See call.

data schema

The format of data for a particular database table.

Device Independent Recording Package (DIRP)

Software that automatically directs data from the various administrative and maintenance facilities to the appropriate recording devices.

Digital Multiplex System (DMS)

A central office switching system in which all external signals are converted to digital data and stored in assigned time slots. Switching is performed by reassigning the original time slots.

directory

In a DMS switch, a software structure that can be used to look up, store, and delete symbols.

directory number (DN)

The full complement of digits required to designate a subscriber's station within one numbering plan area (NPA)-usually a three-digit central office (CO) code followed by a four-digit station number.

DIRP

Device Independent Recording Package

DISKUT

A tool used to access the disk drive units.

DMS

Digital Multiplex System

DMS SuperNode

A central control complex (CCC) for the DMS-100 switch. The two major components of DMS SuperNode are the computing module (CM) and the message switch (MS). Both are compatible with the network module (NM), the input/output controller (IOC), and XMS-based peripheral modules (XPM).

DSKUT

A tool used to access the system load modules (SLM).

EBAF

expanded Bellcore AMA format

expanded Bellcore AMA format (EBAF)

The ability to append additional data in modular form to Bellcore AMA format (BAF) automatic message accounting (AMA) call records. Module codes are used to identify the format of the data appended to the BAF call record. One or more modules can be appended to a BAF call record. *See also* Bellcore AMA format.

input/output controller (IOC)

An equipment shelf that provides an interface between up to 36 I/O devices and the central message controller (CMC). The IOC contains a peripheral processor (PP) that independently performs local tasks, thus relieving the load on the CPU.

integrated services digital network (ISDN)

A set of standards proposed by the CCITT to establish compatibility between the telephone network and various data terminals and devices. ISDN is a communications network that provides access to voice, data, and imaging services from a single type of connector.

IOC

input/output controller

ISDN

Integrated services digital network

log report

A message sent from the DMS switch whenever a significant event has occurred in the switch or one of its peripherals. A log report includes state and activity reports as well as reports on hardware and software faults, test results, and other events or conditions likely to affect the performance of the switch. A log report can be generated in response to a system or manual action.

MAP

Maintenance and administration position. A group of components that provides a user interface between operating company personnel and the DMS-100 Family switches. The interface consists of a visual display unit (VDU) and keyboard, a voice communications module, test facilities, and special furniture.

MAPCI

MAP command interpreter

MAP command interpreter (MAPCI)

A MAP level for accessing maintenance and other functional levels.

message (MSG)

The unit of information transfer between nodes in the DMS-100 switch. A message is incoming if it is sent from a peripheral to the central control (CC) and outgoing if it is sent from the CC to a peripheral. A message is a type of control mechanism used in the I/O messages of the DMS-100 Family switches. The MSG byte specifies that the information to come is a data message.

message type

A descriptor that identifies the function of a message. Stimulus call control has one message type-information. Functional call control has several message types related to call connection, call disconnection, and call status.

MSG

Message

Northern Telecom (NT)

A part of the tricorporate structure consisting of Bell-Northern Research, Bell Canada, and Northern Telecom.

Northern Telecom AMA format (NTFMT)

The format of the output automatic message accounting (AMA) recording data, as specified by Northern Telecom.

Northern Telecom publication (NTP)

A document that contains descriptive information about Northern Telecom hardware or software modules and performance-oriented practices (POP) for installing, testing, or maintaining the system. This document is often supplied as part of the standard documentation package provided to an operating company.

NT

Northern Telecom

NTFMT

Northern Telecom AMA format

NTP

Northern Telecom publication

off-hook

- The condition existing in telephone operations when the receiver or handset is removed from its hookswitch.
- One of two possible signaling states such as tone or no-tone, or ground connection or battery connection.
- The active state (closed loop) of a subscriber or private branch exchange (PBX) line loop.

See also on-hook.

OM

operational measurements

on-hook

- The condition existing in telephone operation when the receiver or handset is resting on its hookswitch.
- One of two possible signaling states, such as tone or no-tone, or ground connection or battery connection.
- The idle state (open loop) of a subscriber or private branch exchange (PBX) line loop.

See also off-hook.

operational measurements (OM)

The hardware and software resources of the DMS-100 Family switches that control the collection and display of measurements taken on an operating system. The OM subsystem organizes the measurement data and manages its transfer to displays and records. The OM data is used for maintenance, traffic, accounting, and provisioning decisions.

peripheral module (PM)

A generic term referring to all hardware modules in the DMS-100 Family switches that provide interfaces with external line, trunk, or service facilities. A PM contains peripheral processors (PP), which perform local routines, thus relieving the load on the CPU.

peripheral module intercept system test (PMIST)

A debugging tool that traces messages between the peripheral modules.

plain old telephone service (POTS)

The basic conventional telephone service. In the context of service screening, POTS is a pseudo-service that is derived from the combination of a bearer service of speech with no supplementary services.

PM

peripheral module

PMIST

peripheral module intercept system test

procedure-oriented type enforcing language (PROTEL)

The high-level programming language used to write software for DMS-100 Family switches.

procedure variable

A variable whose value is the name of a procedure; used by DMS software for dynamic selection of procedures at run time.

process entry module

A module that contains a procedure where a process begins running after initialization.

program

In a DMS switch, a named set of loaded modules that are either named specifically in the definition of a specific program using a loader control file, or implicitly included as part of the program because they are used by other modules in the program.

PROTEL

procedure-oriented type enforcing language

RDW

record descriptor word

real time

The actual time during which the NT40 CPU or DMS-Core SuperNode performs its functions. The time is divided into two main categories: call processing time and noncall processing time.

record descriptor word (RDW)

A 4-byte word that precedes variable-blocked-spanned data records on magnetic tape. Record descriptor word indicates the length of the record.

run time

In a DMS switch, the time during which the CPU is allocated to a process.

schema

The representation of data in a DMS switch as seen by the various users. It includes both tables and associated operations.

SDW

segment descriptor word

segment

The portion of a message in telecommunications that can be contained in a buffer.

segment descriptor word (SDW)

A 4-byte word that identifies segments of variable-length data records on magnetic tape as defined by the record descriptor word.

Service Order System (SERVORD)

A user interface consisting of commands used to change, add, or delete subscriber lines. The format used for commands in the SERVORD comply with the standard telephone industry command format; for example, 3WC is three-way calling, ADO is add option, DEL is delete, and CWT is call waiting.

SERVORD

Service Order system

SLM

system load module

SMDR

Station Message Detail Recording

SOS

Support Operating System

Station Message Detail Recording (SMDR) system

In Meridian Digital Centrex (MDC), a system that provides recording facilities for the details of billable and nonbillable calls for each MDC customer group.

structure code

A numerical value used to designate the type of structure used for a call record.

Support Operating System (SOS)

The software that sets up the environment for loading and executing the application software in the DMS-100 Family system. SOS includes the nucleus, file system, command interpreter, and loader.

system load module (SLM)

A mass storage system in a DMS SuperNode processor that stores office images. From the SLM, new loads or stored images can be booted into the computing module (CM).

table

Two-dimensional entities in which data associated with hardware and software in the DMS-100 Family switches are stored.

terminal

- The point of origination or termination in a communications network.
- Any device capable of sending information, receiving information, or both over a communication channel.
- In a DMS switch, the smallest unit of address space within the input/output (I/O) system.

terminal ID (TID)

In DMS software, the TID uniquely identifies anything on which a call can be originated or terminated.

TID

terminal ID

TIMERDMP

A field in table CRSFMT.

TIMERINT

A field in table CRSFMT.

tuple

The horizontal row of a table.

user

A person, group, or organization who uses the services of a DMS switch.

DMS-100 Family

Automatic Message Accounting (AMA) Tools User Guide

Technical Assistance Manual

Information is subject to change without notice. Northern Telecom reserves the right to make changes in design or components as progress in engineering and manufacturing may warrant.

DMS, DMS-100, DMS SuperNode, MAP, and NT are trademarks of Northern Telecom.

Publication number: TAM-1001-017
Product release: BCS35 and up
Document release: Standard 01.01
Date: March 1993

Printed in the United States of America

northern
telecom