

Digital SNA Application Programming Interface for OpenVMS

Problem Solving

Part Number: AA-NG05C-TE

May 1996

This document tells users of the Digital SNA Application Programming Interface for OpenVMS how to solve problems that could arise in the day-to-day operation of this product.

Revision/Update Information: This is a revised manual.

Operating System and Version: OpenVMS VAX Versions 6.1, 6.2, or 7.0
OpenVMS Alpha Versions 6.1, 6.2, or 7.0

Software Version: Digital SNA Application Programming Interface for OpenVMS, Version 2.4

May 1996

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation or EDS. Digital Equipment Corporation or EDS assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Digital conducts its business in a manner that conserves the environment.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Copyright © 1989, 1996 Digital Equipment Corporation, EDS Defense Limited
All Rights Reserved.

The following are trademarks of Digital Equipment Corporation:

Alpha, DEC, DEC/CMS, DEC/MSS, DECnet, DECSYSTEM-10, DECSYSTEM-20, DECUS, DECwriter, DIBOL, EduSystem, IAS, MASSBUS, OpenVMS, PDP, PDT, RSTS, RSX, UNIBUS, VAX, VAXcluster, VMS, VT, and the Digital logo.

IBM is a registered trademark of International Business Machines Corporation.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface	v
1 Solving API Problems	
1.1 Problem 1: Compile Time Errors	1-2
1.2 Problem 2: Undefined Symbols in Application Program	1-3
1.3 Problem 3: Invalid Parameters in Application Program	1-4
1.4 Problem 4: Faulty Parameter Contents	1-5
1.5 Problem 5: Other Status Messages	1-6
1.6 Problem 6: Application Hangs	1-7
1.7 Problem 7: Protocol Problem	1-8
1.8 Problem 8: BUGCHECK Error Occurs When Running Applications	1-25
Examples	
1-1 Detecting a Bad Sequence Number Using an SNATRACE Physical Unit Trace	1-9
1-2 Detecting an Illegal RU Size Using an SNATRACE Session Trace	1-11
1-3 Detecting a Bracket Protocol Error Using an SNATRACE Session Trace	1-15
1-4 Detecting a Bracket Protocol Error Using a CTF Session Trace	1-20
1-5 Detecting a Bad Sequence Number Using a CTF Trace	1-23

Tables

1-1	API Problems	1-1
-----	--------------------	-----

Preface

The Digital SNA Application Programming Interface (API) for OpenVMS is a Digital Equipment Corporation software product that enables OpenVMS users to communicate with programs running on IBM systems through one of Digital's interconnect systems.

Manual Objectives

This manual provides information to help solve problems that could arise during the day-to-day operation of the API. The material in this guide reflects the experience of product developers. The solutions are not necessarily the only ones that will solve a particular problem, and not every possible problem is discussed.

Intended Audience

This manual is designed to help system managers solve problems with the API. Readers of this manual must have a working knowledge of the Digital and IBM systems that they will be using.

Document Structure

This document consists of one chapter, Solving API Problems, which discusses the problems that may occur with this product and possible solutions to those problems. Each page discusses a different problem. Check the list at the beginning of the chapter to find a problem similar to your own.

Associated Documents

The following is a list of documents related to the Application Programming Interface:

- *Digital SNA Application Programming Interface for OpenVMS Installation*
- *Digital SNA Application Programming Interface for OpenVMS Problem Solving*

- *Digital SNA Application Programming Interface for OpenVMS Programming Programming)*

You should have the following Digital documents available for reference when you use the Application Programming Interface:

- *Digital SNA Domain Gateway Installation*
- *Digital SNA Domain Gateway Management*
- *Digital SNA Domain Gateway Guide to IBM Resource Definition*
- *DECnet SNA Gateway Problem Determination Guide*
- *DECnet SNA Gateway-CT Installation*
- *DECnet SNA Gateway-CT Problem Solving (OpenVMS & ULTRIX)*
- *DECnet SNA Gateway-CT Management (OpenVMS)*
- *DECnet SNA Gateway-CT Guide to IBM Parameters*
- *DECnet SNA Gateway-ST Installation*
- *DECnet SNA Gateway-ST Problem Solving (OpenVMS)*
- *DECnet SNA Gateway-ST Guide to IBM Parameters*
- *DECnet SNA Gateway Management for OpenVMS*
- *Digital Peer Server Installation and Configuration*
- *Digital Peer Server Management*
- *Digital Peer Server Network Control Language Reference*
- *Digital Peer Server Guide to IBM Resource Definition*
- *Digital Peer Server Guide to IBM Resource Definition*
- *OpenVMS SNA Installation*
- *OpenVMS SNA Problem Solving*
- *OpenVMS SNA Guide to IBM Parameters*
- *OpenVMS SNA Management*
- *OpenVMS SNA Problem Determination Guide*

See the following documents for more information about the IBM 3270 Information Display System:

- *ACF for VTAM Version 2, Messages and Codes (IBM Order No. SC27-0614)*

- *IBM 3270 Information Display System and 3274 Control Unit Description and Programmer's Guide* (IBM Order No. GA23-0061)
- *IBM 3287 Printer Models 1 and 2 Component Description* (IBM Order No. GA27-3153)
- *MVS/TSO/VTAM Data Set Print Program Description/Operations Manual* (IBM Order No. SB21-2070)
- *IBM 3270 Information Display System*, Order No. GA23-0060
- *IBM 3270 Information Display System Data Stream Programmer's Reference*, Order No. GA23-0059
- *Systems Network Architecture—Introduction to Sessions Between Logical Units*, Order No. GC20-1869
- *Systems Network Architecture—Sessions Between Logical Units*, Order No. GC20-1868
- *IBM 3270 Information Display System: Operator's Guide*, Order No. GA27-2742

Conventions Used in This Manual

The following table summarizes the graphic conventions that are used in this manual.

Convention	Meaning
Special type	This special type indicates an example of system output or user input.
UPPERCASE	Uppercase letters in command syntax indicate keywords that you can enter. You can enter keywords in either uppercase or lowercase.
<i>italics</i>	Italics in command syntax or examples indicate variables for which either you or the system supplies a value.
[]	Square brackets in command syntax statements indicate that the enclosed value(s) are optional. Default values apply for unspecified options. (Do not type the brackets.)
{ }	Braces in command syntax statements indicate that you must specify one, and only one, of the enclosed values. (Do not type the braces.)
()	Parentheses enclose a group of values that you must specify for an operand. Type the values in the line of code in the order indicated. Type parentheses wherever they appear in a line of code.
<i>hh:mm:ss</i>	Indicates hours, minutes, and seconds
KEY	This symbol indicates a key that you should press. For example, RET means that you should press the RETURN key.
CTRL/x	This notation means that you should hold down the CTRL key and press the key specified by <i>x</i> at the same time.

You must press **RETURN** to enter all commands.

1

Solving API Problems

This chapter discusses problems that may occur when you are using the Digital SNA Application Programming Interface (API) for OpenVMS. Each page presents a problem, and possible solutions for the problem. Table 1-1 lists the API problems discussed in this chapter.

Table 1-1 API Problems

Problem Number	Symptom
1	You get compile time errors with your API program.
2	The linker indicates that the API symbols in your program are not defined.
3	You receive an error status stating that a particular parameter is invalid.
4	You receive an error status stating that the contents of a particular parameter are faulty.
5	Other API status messages are reported.
6	Your application hangs (for example, it enters either the LEF or MWAIT state.)
7	No errors are reported, but nothing works as you expect in the Basic API.
8	You receive a BUGCHECK error while your application is running.

1.1 Problem 1: Compile Time Errors

Problem:

You get compile time errors with your API program.

Solution:

This problem could result from one of the following syntax errors:

- Spelling errors in your code
- Incorrect use of definitions

Make sure you include one of the following definition files from the directory SYSSLIBRARY:

```
SNALIBDEF.ADA
SNALIBDEF.BAS
SNALIBDEF.FOR
SNALIBDEF.H
SNALIBDEF.LIB
SNALIBDEF.MAR
SNALIBDEF.PAS1
SNALIBDEF.PEN1
SNALIBDEF.PLI
SNALIBDEF.R32
SNALU0DEF.ADA
SNALU0DEF.BAS
SNALU0DEF.FOR
SNALU0DEF.H
SNALU0DEF.LIB
SNALU0DEF.MAR
SNALU0DEF.PAS
SNALU0DEF.PEN1
SNALU0DEF.PLI
SNALU0DEF.R32
```

The method for including the definition file is language dependent. For the exact syntax required, see the programmer's guide for the language that you are using.

¹ These files must be created on the system after installation. See the Post-Installation Procedure in the *Digital SNA Application Programming Interface for OpenVMS Installation* manual.

1.2 Problem 2: Undefined Symbols in Application Program

Problem:

The linker indicates that the API symbols in your program are not defined.

Solution:

Make sure that you have linked to the API shareable image section. The following command sequences produce the proper results.

```
$ LINK/MAP test,SY$INPUT/OPTIONS   
SY$SHARE:SNALUOSHR/SHARE   

```

where *test* is the name of your program.

1.3 Problem 3: Invalid Parameters in Application Program

Problem:

You receive an error status stating that a particular parameter is invalid.

Solution:

An invalid parameter error status can indicate that you supplied a parameter in an incorrect form. For example, this could result from passing the parameter by descriptor when you meant to pass it by reference. Check your code.

If you cannot find the problem, one possible solution is to link your program with a debugger. Set a breakpoint at the entry to the API routine you think may be causing the problem. The following text illustrates this process.

If, for example, `SNALU0$REQUEST_CONNECT` is the routine that is causing problems, enter the following commands after you invoke the debugger:

```
DBG> SET LANGUAGE BLISS
DBG> SET BREAK SNALU0$REQUEST_CONNECT
```

When you reach the breakpoint, examine the arguments passed to the routine in question, `SNALU0$REQUEST_CONNECT`. Enter the following command:

```
DBG> EXAMINE .AP: .AP+20
```

This prints the first eight parameters in the argument block. The first longword of the argument block contains the number of parameters that were passed to the `SNALU0$REQUEST_CONNECT` routine. The second and subsequent longwords contain the first and subsequent parameters passed to the `SNALU0$REQUEST_CONNECT` routine. Examine these parameters to see whether you can locate the problem.

For information about the symbolic debugger, refer to the *OpenVMS Debugger Reference Manual*, previously entitled the *OpenVMS Symbolic Debugger Reference Manual*.

Note

The solution to Problem 3 only works on OpenVMS VAX systems.

1.4 Problem 4: Faulty Parameter Contents

Problem:

You receive an error status stating that the contents of a particular parameter are faulty.

Solution:

See the explanation of the particular error status in the *Digital SNA Application Programming Interface for OpenVMS Programming* manual.

1.5 Problem 5: Other Status Messages

Problem:

Other API status messages are reported.

Solution:

The following example describes how to interpret a typical status message reported by the API.

An application using the LU type 2 protocol (which uses the half-duplex flip-flop mode) with CICS reports the following error:

```
ERROR EXECUTING SNALU0$TRANSMIT MESSAGE
STATUS = %SNALU0-E-XMTFAIL, call to SNALU0$TRANSMIT[ W] failed
        -SNA-E-MUTOSENDCHK, MU generated a send check, sense code %X'4009'
```

This error occurred because both the *end-bracket* and *more-data* parameters were set true in a call to SNALU0\$TRANSMIT_MESSAGE. This required the API to set the end bracket indicator (EBI) in an RU that was not the end of the chain. This is a protocol violation described in the IBM manual *Systems Network Architecture Reference Summary* (IBM Order No. GA27-3136) as sense code 4009.

4009 CD Not Allowed: You specified the change direction indicator (CDI) incorrectly (for example, CDI=CD with ECI=-EC, or CDI=CD with EBI=EB).

In this example, the EBI=EB and the ECI=-EC; that is, the EBI equals true and the end chain indicator (ECI) equals false. This is identical to the SNALU0\$TRANSMIT_MESSAGE example just described, since in half-duplex flip-flop mode, EBI equals true is essentially equivalent to CDI equals true.

For further information about status messages, see the explanation of the particular error status in the *Digital SNA Application Programming Interface for OpenVMS Programming* manual.

1.6 Problem 6: Application Hangs

Problem:

Your application hangs; that is, it enters either the LEF or MWAIT state.

Solution:

Check the size of the ASTLM, BIOLM, BYTLM, and DIOLM process quotas. They may be too small.

1.7 Problem 7: Protocol Problem

Problem:

No errors are reported, but nothing works as you expect in the Basic API.

Solution:

Run SNATRACE or CTF Trace to try to locate the protocol problem that exists between your Digital application and the IBM application to which you want to connect.

Note

This problem contains examples of traces output from SNATRACE and CTF trace. The solution to this problem points out particular features of each trace in the notes that follow the trace. In addition, after each trace, this solution notes the steps taken to solve the problem.

Use CTF trace for Digital SNA Domain Gateway and Digital SNA Peer Server. Use SNATRACE for DECnet SNA Gateway-CT, and DECnet SNA Gateway-ST.

The trace examples in this section, can show causes for the following problems:

- **Bad sequence number on positive response.** Example 1-1 and Example 1-4 displays a trace of a problem caused when an OpenVMS application that used the Basic API sent the IBM application a positive response with a bad sequence number. Explanatory notes follow the example.
- **RU size too big.** Example 1-2 displays a trace of a problem caused when an IBM application tried to send the OpenVMS application using the Basic API and an RU that was too big. Explanatory notes follow the example.
- **Bracket protocol violation.** Example 1-3 and Example 1-5 displays a trace of a problem caused by an OpenVMS program using the Basic API that violates SNA bracket protocol. Explanatory notes follow the example.

Example 1-1 Detecting a Bad Sequence Number Using an SNATRACE Physical Unit Trace

```

SNATRACE Version 1.0   Physical Unit Trace  2-NOV-1987 09:11:10.61
Gateway node ARWEN    Circuit SNA-0
(Protocol version = 1.0.0, Buffering level = 3, Data size = 272)

T TH=2C0000090000 RH=0B8000 RU=21. bytes      (012,022,000001)  1
  FID2,OS,DAF=00,OAF=09,SNF=0000
  RQ,FMD,FI,BCI,ECI,DR1I
  0106 8100 4040 4040 4040 4040 F304 C3C9 :  \a.          3\CI
  C3E2 0000 00                               : CS...

R TH=2C0009000000 RH=8B8000 RU=3. bytes      (014,000,000001)  2
  FID2,OS,DAF=09,OAF=00,SNF=0000
  +RSP,FMD,FI,BCI,ECI,DR1I
  0106 81                                     :  \a

R TH=2D0009010162 RH=6B8000 RU=33. bytes    (014,000,000001)  3
  FID2,OS,EFI,DAF=09,OAF=01,SNF=0162
  RQ,SC,FI,BCI,ECI,DR1I
  3101 0303 B1B0 3080 0004 8585 0400 0200 :  \...\\\.\ee\...
  0000 0000 0000 0000 0000 0004 C3C9 C3E2 :  ..... \CICS
  00                                         :  .

T TH=2D0001090162 RH=EB8000 RU=1. byte      (012,002,000001)  4
  FID2,OS,EFI,DAF=01,OAF=09,SNF=0162
  +RSP,SC,FI,BCI,ECI,DR1I
  31                                         :  \

R TH=2D0009010163 RH=6B8000 RU=1. byte      (014,000,000001)  5
  FID2,OS,EFI,DAF=09,OAF=01,SNF=0163
  RQ,SC,FI,BCI,ECI,DR1I
  A0                                         :  \

T TH=2D0001090000 RH=EB8000 RU=1. byte      (012,010,000001)  6
  FID2,OS,EFI,DAF=01,OAF=09,SNF=0000
  +RSP,SC,FI,BCI,ECI,DR1I
  A0                                         :  \

```

The following notes correspond to messages recorded in the trace and are keyed to the numeric references in Example 1-1.

- 1 The DECnet SNA Gateway sends out an INIT-SELF request to IBM SSCP, requesting a connection with the IBM application CICS.
- 2 The SNA Gateway receives a positive response from the IBM SSCP for the INIT-SELF request.
- 3 The IBM application CICS sends a BIND request to the DECnet SNA Gateway for logical unit 9.

- 4 The API program sends a positive response to the BIND request received from the IBM application.
- 5 The IBM application, CICS, sends an SDT message to the Digital SNA OpenVMS application program.
- 6 The OpenVMS application program sends a positive response to the SDT message received from the IBM application.

Note

The session hangs at this point and does not pass any data. This occurs because the sequence number of the last response is incorrect: the sequence number in the SDT was X'0163' and the response is numbered X'0000' (they should have identical numbers).

When a problem like this occurs, the IBM application finds no protocol errors. The IBM application ignores the response because it does not have any outstanding requests with the same sequence number. The IBM application does not report an error; rather, it waits until the correct response comes in, thus preventing any additional data traffic. (It is not possible for the IBM application to send messages about the bad response, since the SNA protocol does not allow a response to a response.)

To prevent the session from hanging at this point, the API programmer must alter the API program to make it return the correct sequence number in its response to the IBM application.

Example 1-2 Detecting an Illegal RU Size Using an SNATRACE Session Trace

```

SNATRACE Version 1.0   Session Trace   2-NOV-1987 10:07:44.51
Gateway node ARWEN    Circuit SNA-0   Session 1
(Protocol version = 1.0.0, Buffering level = 3, Data size = 272)

T TH=2C0000010000 RH=0B8000 RU=22. bytes   (012,022,000001)  1
  FID2,OS,DAF=00,OAF=01,SNF=0000
  RQ,FMD,FI,BCI,ECI,DR1I
  0106 8100 C4E2 C9D3 C7D4 D6C4 F305 D5C3 : .\a.DSILGMOD3.NC
  C3C6 F200 0000                               : CF2...

R TH=2C0001000000 RH=8B8000 RU=3. bytes   (014,000,000001)  2
  FID2,OS,DAF=01,OAF=00,SNF=0000
  +RSP,FMD,FI,BCI,ECI,DR1I
  0106 81                                       : .\a

R TH=2D0001010168 RH=6B8000 RU=34. bytes  (014,000,000001)  3
  FID2,OS,EFI,DAF=01,OAF=01,SNF=0168
  RQ,SC,FI,BCI,ECI,DR1I
  3101 0303 B1A0 3080 0004 8585 0000 0200 : \...\...\ee....
  0000 0000 1850 1850 7F00 0005 D5C3 C3C6 : .....&.&"...NCCF
  F200                                         : 2.

T TH=2D0001010168 RH=EB8000 RU=1. byte   (012,002,000001)  4
  FID2,OS,EFI,DAF=01,OAF=01,SNF=0168
  +RSP,SC,FI,BCI,ECI,DR1I
  31                                             : \

R TH=2D0001010169 RH=6B8000 RU=2. bytes  (014,000,000001)  5
  FID2,OS,EFI,DAF=01,OAF=01,SNF=0169
  RQ,SC,FI,BCI,ECI,DR1I
  3202                                         : ..

T TH=2D0001010169 RH=EB8000 RU=1. byte   (012,016,000001)  6
  FID2,OS,EFI,DAF=01,OAF=01,SNF=0169
  +RSP,SC,FI,BCI,ECI,DR1I
  32                                             : .

R TH=2D000101016B RH=6B8000 RU=37. bytes  (014,000,000001)  7
  FID2,OS,EFI,DAF=01,OAF=01,SNF=016B
  RQ,SC,FI,BCI,ECI,DR1I
  3101 0303 B1A0 3080 0004 8585 0000 0200 : \...\...\ee....
  0000 0000 1850 1850 7F00 0008 D5C3 C3C6 : .....&.&"...\NCCF
  F2F0 F0F0 00                               : 2000.

```

(continued on next page)

Example 1–2 (Cont.) Detecting an Illegal RU Size Using an SNATRACE Session Trace

```

T TH=2D000101016B RH=EB8000 RU=1. byte      (012,002,000001)  8
  FID2,OS,EFI,DAF=01,OAF=01,SNF=016B
  +RSP,SC,FI,BCI,ECI,DR1I
  31                                           : \
R TH=2D000101016C RH=6B8000 RU=1. byte      (014,000,000001)  9
  FID2,OS,EFI,DAF=01,OAF=01,SNF=016C
  RQ,SC,FI,BCI,ECI,DR1I
  A0                                           : \
T TH=2D000101016C RH=EB8000 RU=1. byte      (012,010,000001) 10
  FID2,OS,EFI,DAF=01,OAF=01,SNF=016C
  +RSP,SC,FI,BCI,ECI,DR1I
  A0                                           : \
R TH=280001010001 RH=0381C0 RU=209. bytes   (014,000,000001) 11
  FID2,FS,DAF=01,OAF=01,SNF=0001
  RQ,FMD,BCI,ECI,DR1I,PI,BBI,EBI
  F540 11C1 5CD5 D5D5 11C1 E5D5 D511 C16D : 5 .A*NNN.AVNN.A
  3CC1 F4C3 11C1 7B3C C2C2 C311 C2C7 3CC2 : .A4C.A#.BBC.BG.B
  D2C6 11C2 6CD5 D5D5 D511 C2F5 D5D5 11C2 : KF.B%NNNN.B5NN.B
  7CC3 C3C3 4040 40C3 C3C3 11C3 4AC3 C3C3 : @CCC CCC.C[CCC
  4040 40C3 C3C3 11C3 D7C6 C611 C37C 3CC4 : CCC.CPFF.C@.D
  C1D5 4040 4040 D5D5 4040 4040 C3C3 C311 : AN NN CCC.
  C4D9 C3C3 C311 C4E7 C6C6 11C5 4CD5 D540 : DRCCC.DXFF.E<NN
  D5D5 D540 4040 D5D5 11C5 5BC3 C311 C5E9 : NNN NN.E$CC.EZ
  C3C3 11C5 F7C6 C611 C65C D5D5 4040 D5D5 : CC.E7FF.F*NN NN
  D540 40D5 D511 C66B C3C3 11C6 F9C3 C311 : N NN.F,CC.F9CC.
  C7C7 3CC7 4EC6 11C7 6CD5 D540 4040 D5D5 : GG.G+F.G%NN NN
  D540 D5D5 11C7 7BC3 C311 C8C9 C3C3 11C8 : N NN.G#CC.HICC.H
  D7C6 C611 C87C D5D5 11C9 C23C C9C7 D511 : PFF.H@NN.IB.IGN.
  C9                                           : I
R TH=240001010001 RH=none RU=92. bytes      (014,000,000001) 12
  FID2,LS,DAF=01,OAF=01,SNF=0001
  4BC3 C3C3 11C9 D9C3 C3C3 11C9 E7C6 C611 : .CCC.IRCCC.IXFF.
  4A4C D5D5 114A D33C 4AD7 D511 4A5C C3C3 : [<NN.[L.[PN.[*CC
  C340 4040 C3C3 C311 4A6A C3C3 C340 4040 : C CCC.[|CCC
  C3C3 C311 4AF7 C6C6 114B 5CD5 D511 4BE4 : CCC.[7FF.*NN..U
  D5D5 D511 4B6D 3C4B F4C3 114B 7B3C 4CC2 : NNN...4C.##.<B
  C311 4CC7 C6C6 11C1 4D40 4040           : C.<GFF.A(
T TH=2C0001010001 RH=879100 RU=4. bytes     (012,000,000001) 13
  FID2,OS,DAF=01,OAF=01,SNF=0001
  -RSP,FMD,SDI,BCI,ECI,DR1I,PI
  8010 0000                                   : \...

```

The following notes correspond to messages recorded in the trace and are keyed to the numeric references in Example 1–2.

- 1 The Digital SNA Gateway sends an INIT-SELF request message to the IBM SSCP, requesting a connection to the IBM application NCCF2.
- 2 The SNA Gateway receives a positive response from the IBM SSCP for the INIT-SELF request message.
- 3 The IBM application NCCF2 sends a BIND request to the Digital SNA Gateway for a logical unit.
- 4 The SNA Gateway sends a positive response for the BIND request.
- 5 The SNA Gateway receives an UNBIND, BIND forthcoming message from IBM application NCCF2. (New IBM address space has been allocated.)
- 6 The SNA Gateway sends a positive response for the UNBIND message.
- 7 The SNA Gateway receives a new BIND request from the IBM application NCCF2000.
- 8 The SNA Gateway sends a positive response for the BIND request.
- 9 The IBM application NCCF2000 sends an SDT message to the SNA Gateway.
- 10 The API program sends a positive response for the SDT message.
- 11 The IBM application sends the first segment of an RU for the Digital SNA OpenVMS application program.
- 12 The IBM application sends the last segment of an RU for the Digital SNA OpenVMS application program.
- 13 The SNA Gateway sends a negative response to the IBM host for message 11. (The API program does not receive an error report.)

Note

Since the contracted RU size from the BIND is 256 bytes (BIND parameter byte 11 is X'85'), the Gateway's physical unit services have detected an error in segmenting: the RU size in message 11 is 209 bytes and the RU size in message 12 is 92 bytes, for a total exceeding 256 bytes.

To solve the problem, the IBM system programmer should change the VTAM logon mode table entry. In this example, DSILGMOD is changed from a value of X'85' to X'87'.

Example 1-3 Detecting a Bracket Protocol Error Using an SNATRACE Session Trace

```

SNATRACE Version 1.0   Session Trace   2-NOV-1987 12:01:15.09
Gateway node SNARK    Circuit SNA-0   Session 9
(Protocol version = 1.0.0, Buffering level = 3, Data size = 48)

T TH=2C0000090000 RH=0B8000 RU=21. bytes      (012,022,000001)  1
  FID2,OS,DAF=00,OAF=09,SNF=0000
  RQ,FMD,FI,BCI,ECI,DR1I
  0106 8100 4040 4040 4040 4040 F304 C3C9 :  \.a.          3\CI
  C3E2 0000 00                               :  CS...

R TH=2C0009000000 RH=8B8000 RU=3. bytes      (014,000,000001)  2
  FID2,OS,DAF=09,OAF=00,SNF=0000
  +RSP,FMD,FI,BCI,ECI,DR1I
  0106 81                                     :  \.a

R TH=2D00009010039 RH=6B8000 RU=33. bytes   (014,000,000001)  3
  FID2,OS,EFI,DAF=09,OAF=01,SNF=0039
  RQ,SC,FI,BCI,ECI,DR1I
  3101 0303 B1B0 3080 0004 8585 0400 0200 :  \...\...\ee\...
  0000 0000 0000 0000 0000 0004 C3C9 C3E2 :  .....CICS
  00                                         :  .

T TH=2D0001090039 RH=EB8000 RU=1. byte      (012,002,000001)  4
  FID2,OS,EFI,DAF=01,OAF=09,SNF=0039
  +RSP,SC,FI,BCI,ECI,DR1I
  31                                         :  \

R TH=2D000901003A RH=6B8000 RU=1. byte      (014,000,000001)  5
  FID2,OS,EFI,DAF=09,OAF=01,SNF=003A
  RQ,SC,FI,BCI,ECI,DR1I
  A0                                         :  \

T TH=2D000109003A RH=EB8000 RU=1. byte      (012,010,000001)  6
  FID2,OS,EFI,DAF=01,OAF=09,SNF=003A
  +RSP,SC,FI,BCI,ECI,DR1I
  A0                                         :  \

R TH=2C0009010001 RH=4B8100 RU=1. byte      (014,000,000001)  7
  FID2,OS,DAF=09,OAF=01,SNF=0001
  RQ,DFC,FI,BCI,ECI,DR1I,PI
  C8                                         :  H

```

(continued on next page)

Example 1-3 (Cont.) Detecting a Bracket Protocol Error Using an SNATRACE Session Trace

```

T TH=2C0001090000 RH=830100 RU=0. bytes      (012,022,000001)  8
  FID2,OS,DAF=01,OAF=09,SNF=0000
  +RSP,FMD,BCI,ECI,PI
T TH=2C0001090001 RH=CB8000 RU=1. byte      (012,006,000001)  9
  FID2,OS,DAF=01,OAF=09,SNF=0001
  +RSP,DFC,FI,BCI,ECI,DR1I
  C8                                     : H
R TH=2C0009010002 RH=0290C0 RU=256. bytes  * (014,000,000001) 10
  FID2,OS,DAF=09,OAF=01,SNF=0002
  RQ,FMD,BCI,DR1I,ERI,BBI,EBI
  F5C7 1DF0 C4C5 C340 E2F3 F7F0 40E2 E4D7 : 5G.0DEC S370 SUP
  D7D6 D9E3 40C6 C1C3 C9D3 C9E3 E840 4040 : PORT FACILITY
  4040 40                                     :
R TH=2C0009010003 RH=009000 RU=256. bytes  * (014,000,000001) 11
  FID2,OS,DAF=09,OAF=01,SNF=0003
  RQ,FMD,DR1I,ERI
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 40                                     :
R TH=2C0009010004 RH=009000 RU=256. bytes  * (014,000,000001) 12
  FID2,OS,DAF=09,OAF=01,SNF=0004
  RQ,FMD,DR1I,ERI
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 40                                     :
R TH=2C0009010005 RH=009100 RU=256. bytes  * (014,000,000001) 13
  FID2,OS,DAF=09,OAF=01,SNF=0005
  RQ,FMD,DR1I,ERI,PI
  E240 4040 4040 4040 4040 E5E5 E540 4040 : S          VVV
  E5E5 E540 4040 E2E2 E2E2 E2E2 E2E2 4040 : VVV   SSSSSSSS
  4040 40                                     :
R TH=2C0009010006 RH=009000 RU=256. bytes  * (014,000,000001) 14
  FID2,OS,DAF=09,OAF=01,SNF=0006
  RQ,FMD,DR1I,ERI
  4040 4040 4040 E2E2 E2E2 4040 4040 4040 :          SSSS
  4040 4040 4040 40C3 C3C3 C340 40C3 C340 :          CCCC  CC
  4040 40                                     :

```

(continued on next page)

Example 1–3 (Cont.) Detecting a Bracket Protocol Error Using an SNATRACE Session Trace

```

R TH=2C0009010007 RH=009000 RU=256. bytes * (014,000,000001) 15
  FID2,OS,DAF=09,OAF=01,SNF=0007
  RQ,FMD,DR1I,ERI
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 40 :
R TH=2C0009010008 RH=009000 RU=256. bytes * (014,000,000001) 16
  FID2,OS,DAF=09,OAF=01,SNF=0008
  RQ,FMD,DR1I,ERI
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 40 :
T TH=2C0001090000 RH=830100 RU=0. bytes (012,022,000001) 17
  FID2,OS,DAF=01,OAF=09,SNF=0000
  +RSP,FMD,BCI,ECI,PI
R TH=2C0009010009 RH=019100 RU=131. bytes * (014,000,000001) 18
  FID2,OS,DAF=09,OAF=01,SNF=0009
  RQ,FMD,ECI,DR1I,ERI,PI
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 4040 4040 4040 4040 4040 4040 4040 :
  4040 40 :
T TH=2C0001090000 RH=830100 RU=0. bytes (012,022,000001) 19
  FID2,OS,DAF=01,OAF=09,SNF=0000
  +RSP,FMD,BCI,ECI,PI
T TH=2C0001090001 RH=0380A0 RU=1. byte (012,006,000001) 20
  FID2,OS,DAF=01,OAF=09,SNF=0001
  RQ,FMD,BCI,ECI,DR1I,BBI,CDI
  6D :
R TH=2C0009010001 RH=838000 RU=0. bytes (014,000,000001) 21
  FID2,OS,DAF=09,OAF=01,SNF=0001
  +RSP,FMD,BCI,ECI,DR1I

```

(continued on next page)

Example 1–3 (Cont.) Detecting a Bracket Protocol Error Using an SNATRACE Session Trace

```

R TH=2C000901000A RH=039040 RU=2. bytes      (014,000,000001) 22
  FID2,OS,DAF=09,OAF=01,SNF=000A
  RQ,FMD,BCI,ECI,DR1I,ERI,EBI
  F1C3                                         : 1C
T TH=2C0001090002 RH=038000 RU=15. bytes     (012,006,000001) 23
  FID2,OS,DAF=01,OAF=09,SNF=0002
  RQ,FMD,BCI,ECI,DR1I
  7D40 4CC3 C5D4 E340 C9D5 D840 E3C5 D9     : ' <CEMT INQ TER
R TH=2D000901003B RH=6B8000 RU=2. bytes      (014,000,000001) 24
  FID2,OS,EFI,DAF=09,OAF=01,SNF=003B
  RQ,SC,FI,BCI,ECI,DR1I
  3201                                         : ..
T TH=2D000109003B RH=EB8000 RU=1. byte       (012,016,000001) 25
  FID2,OS,EFI,DAF=01,OAF=09,SNF=003B
  +RSP,SC,FI,BCI,ECI,DR1I
  32                                         : .

```

The following notes correspond to messages recorded in the trace and are keyed to the numeric references in Example 1–3.

- 1 The Digital SNA Gateway sends an INIT-SELF request to the IBM SSCP, requesting a connection with the IBM application CICS.
- 2 The SNA Gateway receives a positive response from the IBM SSCP for the INIT-SELF request.
- 3 The IBM application, CICS, sends a BIND request to the Digital SNA SNA Gateway for logical unit 9. This BIND request specifies use of bracket protocol (IBM's BIND bit 2 of byte 6 is set) in this session.
- 4 The OpenVMS application program (emulating a 3278 terminal) sends a positive response to the BIND request from CICS.
- 5 The IBM application, CICS, sends an SDT message to the Digital SNA OpenVMS application program.
- 6 The OpenVMS application program sends a positive response to the SDT message received from the IBM application CICS.
- 7 CICS sends a BID to request permission to initiate a bracket.
- 8 The Digital SNA Gateway sends an isolated pacing response to CICS.
- 9 The OpenVMS application program sends a positive response to the BID request received from the IBM application CICS.

- 10 CICS sends its logo to the terminal.
- 11 CICS sends its logo to the terminal.
- 12 CICS sends its logo to the terminal.
- 13 CICS sends its logo to the terminal.
- 14 CICS sends its logo to the terminal.
- 15 CICS sends its logo to the terminal.
- 16 CICS sends its logo to the terminal.
- 17 The Digital SNA Gateway sends an isolated pacing response to CICS.
- 18 CICS completes sending its logo to the application program/terminal.
- 19 The OpenVMS application sends a positive response to CICS.
- 20 The OpenVMS application user presses the CLEAR attention identifier (AID) key.
- 21 CICS sends the OpenVMS application a positive response to the CLEAR AID message.
- 22 CICS restores the OpenVMS application user's keyboard and resets the modified data tag (MDT) bits.
- 23 The OpenVMS application sends a message; the begin bracket indicator (BBI) and the change direction indicator (CDI) are not set. (RH is X'038000'; it should be X'0380A0'.)
- 24 CICS sends an UNBIND request, ending the session.
- 25 The OpenVMS application sends a positive response to the UNBIND request.

Note

CICS has ended the session with the OpenVMS application program.

Because the BIND request that the OpenVMS application and CICS agreed to specified that SNA bracket protocol must be followed, CICS detected a protocol error after receiving message 23. When CICS detected the error, it sent the UNBIND request (message 24) to the OpenVMS application.

The OpenVMS application programmer corrects the problem. The programmer ensures that the application sends a message that follows the protocol agreed to in the BIND for that session.

Example 1-4 Detecting a Bracket Protocol Error Using a CTF Session Trace

CTF V1.0-00 Page 1
 Trace started on 18-APR-1996 14:46:15.11 Analyzed on 18-APR-1996 14:54:11.20
 Trace File CTF\$TRACE.DAT;1 Output File CTF\$TRACE.LOG;1

TH	RH	Data
ONA=1.936 DNA=60.3 SNF=0039	REQ SC FI OC DR1	31 01 03 03 B1 90 30 82 04 84 F8 F8 84 04 1 02 00 00 00 00 00 18 50 18 50 7F 00 00 10 D5 C5 E3 E6 D6 D9 D2 F1 4B C3 C9 C3 E2 C2 C9 C7 00 00 11 D5 C5 E3 E6 D6 D9 D2 F1 4B C8 F0 F1 C1 C1 F0 F0 F2 60 17 E3 43 C1 F8 12 82 2F 3A 0E D5 C5 E3 E6 D6 D9 D2 F1 4B C3 D7 F1 F6 F0 2C 0A 01 08 40 40 40 40 40 40 40 40 2D 09 08 E7 E3 D3 C1 D9 C7 C5 40 \...\b\d88d\.....&".NETWORK1.CICSBI G...NETWORK1.H01AA002-\T\A8.b.\.NETWORK1.CP 160\\.\ .\XTLARGE
ONA=60.3 DNA=1.936 SNF=0039	+RSP SC FI OC DR1	31 01 00 00 00 00 80 02 00 00 F8 F8 80 00 2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 17 E3 43 C1 F8 12 82 2F 3A 0E D5 C5 E3 E6 D6 D9 D2 F1 4B C3 D7 F1 F6 F0 \.....\...88\.....-\T\A8.b.\.N ETWORK1.CP160
ONA=1.936 DNA=60.3 SNF=003A ONA=60.3 DNA=1.936 SNF=003A SNF=7DFB	REQ SC FI OC DR1 +RSP SC FI OC DR1	A0 3 \ A0 4 \

(continued on next page)

Example 1-4 (Cont.) Detecting a Bracket Protocol Error Using a CTF Session Trace

```

ONA=1.936   |REQ  DFC FI OC PI   |C8  5
DNA=60.3    |DR1                 |H
SNF=0001
ONA=60.3    |+RSP FMD   OC PI   |  6
DNA=1.936   |
SNF=0000
ONA=60.3    |+RSP DFC FI OC     |C8  7
DNA=1.936   |DR1                 |H
SNF=0001
ONA=1.936   |REQ  FMD   OC      |F5 C7 1D F0 E6 C5 D3 C3 D6 D4 C5 40 E3 D6  8
DNA=60.3    |DR1      ER        |40 C4 C5 C3 7D E2 40 E2 61 F3 F7 F0 40 E2
SNF=0002    |      BB  EB        |E4 D7 D7 D6 D9 E3 40 C6 C1 C3 C9 D3 C9 E3
              |                    |E8 40 60 40 C1 D7 D7 D3 C9 C4 40 C3 C9 C3
              |                    |E2 C2 C9 C7 40 F1 F4 7A F4 F3 7A F5 F1 11
              |                    |C7 61 40 5C 5C 5C 5C 5C 5C E0 40 40 5C 5C
              |                    |5C 5C 5C 5C E0 40 40 5C 5C 5C 5C 5C 5C E0
              |                    |40 40 40 5C 5C 5C 5C 5C 5C E0 40 40 40 40
              |                    |40 40 40 40 40 40 5C E0 40 5C 5C 5C 5C 5C
              |                    |5C 5C 5C E0 40 40 5C 5C 5C 5C 5C 5C E0 40
              |                    |40 40 40 5C 5C 5C 5C E0 40 40 11 C8 F1 5C
              |                    |5C 5C 5C 5C 5C
              |                    |5G.0WELCOME TO DEC'S S/370 SUPPORT FACILITY
              |                    |- APPLID CICSBIG 14:43:51.G/ *****\ ****
              |                    |**\ *****\ *****\ * \ *****
              |                    |\ *****\ *****\ .H1*****
ONA=60.3    |REQ  FMD   OC PI   |6D  9
DNA=1.936   |DR1      ER        |CD -
SNF=0001    |      BB           |
ONA=1.936   |+RSP FMD   OC PI   | 10
DNA=60.3    |
SNF=0001
ONA=1.936   |REQ  FMD   OC      |F5 C3 11
DNA=60.3    |DR1      ER        |5C
SNF=0003    |      EB           |
ONA=60.3    |REQ  FMD   OC      |7D 40 4C C3 C5 D4 E3 40 C9 D5 D8 40 E3 C5 12
DNA=1.936   |DR1      ER        |D9
SNF=0002    |      <CEMT INQ TER
ONA=1.936   |REQ  SC   FI OC     |32 01 00 00 00 00 60 17 E3 43 C1 F8 12 82 13
DNA=60.3    |DR1                 |2F 3A 0E D5 C5 E3 E6 D6 D9 D2 F1 4B C3 D7
SNF=003B    |                    |F1 F6 F0
              |                    |.....-T\A8.b.\.NETWORK1.CP160
ONA=60.3    |+RSP SC   FI OC     |32 14
DNA=1.936   |DR1                 |.
SNF=003B

```

The following notes correspond to the messages recorded in the trace and are keyed to the numeric references in Example 1–4.

- 1 The IBM application CICSBIG sends a BIND request to the Digital SNA Domain Gateway for logical unit H01AA002. This BIND request specifies use of bracket protocol (IBM's BIND bit 2 of byte 6 is set) in this session.
- 2 The OpenVMS application program (emulating a 3278 terminal) sends a positive response to the BIND request from CICSBIG.
- 3 The IBM application, CICSBIG, sends an SDT message to the Digital SNA OpenVMS application program.
- 4 The OpenVMS application program sends a positive response to the SDT message received from the IBM application CICSBIG.
- 5 CICSBIG sends a BID to request permission to initiate a bracket.
- 6 The Digital SNA Gateway sends an isolated pacing response to CICSBIG.
- 7 The OpenVMS application program sends a positive response to the BID request received from the IBM application CICSBIG.
- 8 CICS send its logo to the terminal.
- 9 The OpenVMS application user presses the CLEAR attention identifier (AID) key.
- 10 The Digital SNA Gateway sends an isolated pacing response to CICSBIG.
- 11 CICSBIG restores the OpenVMS application user's keyboard, resets the modified data tag (MDT) bits and erases the screen.
- 12 The OpenVMS applications sends a message; the begin bracket indicator (BBI) and the change direction indicator (CDI) are not set.
- 13 CICSBIG sends an UNBIND request, ending the session.
- 14 The OpenVMS application send a positive response to the UNBIND request.

The following notes correspond to the messages recorded in the trace and are keyed to the numeric references in Example 1–5.

- 1 The IBM application CICSBIG sends a BIND request to the Digital SNA Domain Gateway for logical unit H01AA002.
- 2 The API program sends a positive response to the BIND request received from the IBM application.
- 3 The IBM application, CICSBIG, sends an SDT message to the Digital SNA OpenVMS application program.
- 4 The OpenVMS application program sends a positive response to the SDT message received from the IBM application.

1.8 Problem 8: BUGCHECK Error Occurs When Running Applications

Problem:

You receive a BUGCHECK error while your application is running.

Solution

Check to see that:

- Your application is not writing into memory that it did not declare or allocate.
- You are using dynamic memory correctly; that is, allocating and deallocating memory correctly. If you have deallocated a piece of memory, be sure that you are not still writing to it.

If you are not using dynamic memory, report the problem with a Software Problem Report (SPR). For information on how to report your problem, refer to the *DECnet SNA Gateway Problem Determination Guide* or the *OpenVMS SNA Problem Determination Guide* (OpenVMS VAX Version 6.1 and Version 6.2 only).

