IBM DB2 Connect
APPC, CPI-C, and SNA Sense Codes
Version 5

Document Number SDB2-CONN-AP

IBM DB2 Connect

APPC, CPI-C, and SNA Sense Codes

Version 5

SDB2-CONN-AP

IBM DB2 Connect

APPC, CPI-C, and SNA Sense Codes

Version 5

Before using this information and the product it supports, be sure to read the general information under Appendix A, “Notices” on page 103.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling `1-800-879-2755` in U.S. or `1-800-IBM-4YOU` in Canada.

Contents

About This Book

This book contains general information about APPC, CPI-C, and SNA Sense codes that you may encounter when using the following IBM DB2 products:

- DB2 Connect Personal Edition (formerly known as DDCS Single-User).
- DB2 Connect Enterprise Edition, for AIX, HP-UX, Solaris, OS/2, and Windows NT (formerly known as DDCS Multi-user Gateway).
- DB2 Universal Database Enterprise Edition, for AIX, HP-UX, Solaris, OS/2, and Windows NT.
- DB2 Universal Database Workgroup Edition, for AIX, HP-UX, Solaris, OS/2, and Windows NT.

This book is primarily intended for programmers and administrators who are responsible for setting up and maintaining DB2 Connect connections between DB2 Universal Database clients and any of the following DRDA application server database management systems:

- DATABASE 2 for OS/390 (DB2 for OS/390) Version 5
- DATABASE 2 for MVS (DB2 for MVS) Version 3 or higher
- DATABASE 2 for VSE and VM (DB2 for VSE & VM)
- DATABASE 2 for AS/400 (DB2 for AS/400)
- Any other relational database management system that implements Distributed Relational Database Architecture (DRDA) application server function.

Notes:

1. DB2 Universal Database Version 5.0 does not require DB2 Connect in order to function as a DRDA application server.
2. DB2 for OS/390 Version 5.1 or higher is required in order to use DRDA Level 3 functions, including TCP/IP database connections, and stored procedures with multi-row answer sets.

DB2 Connect is part of the DB2 Universal Database family of products.

How This Book is Structured

This book contains the following major sections:

- Chapter 1, “CPI-C Return Codes: Introduction” on page 1
- Chapter 2, “APPC/APPN Sense Data: Introduction” on page 37.

Chapter 1. CPI-C Return Codes: Introduction

Every CPI-C call has a return code as its last parameter. The return code reports back from CPI-C on what happened on a call. Return codes don't just report success or failure; sometimes they report the arrival of a message from the partner, such as when the partner has issued a Send_Error() call.

When using the APPC protocol and you encounter communications error code 30081, the error tokens for the 30081 message list the CPI-C function that returned the error, and the return code from the CPI-C function call.

The following is a description for each CPI-C return code. The return codes are listed in numerical order, since many CPI-C programs and utilities display or log the return code number and not its accompanying name.

After all the descriptions is a table that lets you map in the other direction; it lists the CPI-C return code names in alphabetical order, followed by their numerical value in decimal (and hexadecimal).

Note: This information can also be found in Appendix A of the book *CPI-C Programming in C: An Application Developer's Guide to APPC*, published by McGraw-Hill, New York, NY, ISBN 0-07-911733-3.

CPI-C Return Codes (0 to 99)

The following return codes apply to all programs that issue CPI-C calls.

0 CM_OK

A CPI-C call completed successfully.

Explanation

CPI-C executed the function that was requested.

Programmer Action

This is an “expected” return code. No action is required. This return code should be handled as part of the mainline program logic for programs that issue CPI-C calls.

Operator Action

None.

1 CM_ALLOCATE_FAILURE_NO_RETRY

A session cannot be obtained for a conversation, and human intervention will be needed to correct the problem.

Explanation

The local program issued an Allocate() call (CMALLC), but there was difficulty in activating a link or a session.

CPI-C replies with this return code when it has determined that it is impossible to allocate a session without some human intervention. For example, the local communications software can tell when there is no adapter in the local computer, making a conversation impossible.

If there is any hope that a conversation might be established, CPI-C responds with the CM_ALLOCATE_FAILURE_RETRY return code. For example, CPI-C cannot look in the remote computer and see if there is a communications adapter there or see if it is even powered on. CPI-C retains some hope that a conversation might still be established by simply retrying the Allocate() call.

When you get this return code in a production environment, the problem is frequently a transient link failure—which is recoverable. Retrying the Allocate will generally be successful. Contrast this with a test environment, where the conversation is being established for the first time. In a test environment, either return code is likely, and there is probably a setup problem with the network hardware or software.

The implication of this return code is that something is wrong at the local computer, but this is not necessarily always true. (Experience suggests that for any of these problems, it is likely both computers are set up wrong, as is the connection between them!)

This conversation with the partner is over.

Likely Causes

There are thousands of reasons for this return code. The local computer is unable to set up the session it needs with the remote computer. Here are some likely causes:

- The partner LU name is not really in the network or cannot be reached.
- The partner LU name might be incorrect or the target computer is not active or is not reachable. Partner LU names are configured as part of the CPI-C side information, or set with a call to a Set_Partner_LU_Name().
- The mode name requested for this session is not configured or spelled correctly at the partner.
- There is no route through the network that satisfies the requirements of the mode name. For example, there isn't a secure route that meets the requirements associated with #INTERSC.

Programmer Action

There are thousands of reasons for this return code. A specific reason for the failure can be found in the *SNA sense data* associated with this return code. To diagnose the problem, your program must provide this SNA sense data to its user, or indicate a way it can be obtained. You should consider directing your users to find sense data values in traces and error logs.

The local program should not try to allocate a session again until the condition is corrected.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Examine any platform-related message logs and error logs at both locations to find more information about this problem. Look for equipment failures or setup problems related to the network components and the computers using them. Fix the problem, as indicated by any SNA sense data associated with this call. You may need to run the applications again with SNA tracing activated to get the sense data.

After correcting the problem, try running the pair of programs again, if appropriate.

2 CM_ALLOCATE_FAILURE_RETRY

A session cannot be obtained for a conversation; the problem may be temporary, but human intervention will probably be needed to correct the problem.

Explanation

The local program issued an Allocate() call (CMALLC), but the underlying network software was unable to obtain or activate a session.

CPI-C replies with this return code when it has determined that it cannot allocate a session for a conversation. If the Allocate() request is tried again at a later time, the session might be able to be obtained, without human intervention. For example, CPI-C cannot look in the remote computer and see if there is a matching communications adapter there. CPI-C retains some hope that a conversation might still be established by simply retrying the Allocate() call.

Although this return code has the word “RETRY” in it, the allocation request might never succeed, especially if the partner is never activated or has been configured wrong.

This conversation with the partner is over.

Likely Causes

There are thousands of reasons for this return code. The local computer is unable to set up the session it needs with the remote computer. Here are some likely causes:

- The remote computer is not powered on or has no communications adapter.
- The underlying CPI-C or APPC software on the remote computer is stopped, has not been started or is not yet fully active, or is abended.
- The DLC at the remote computer is not configured correctly.
- One computer, using a dependent LU 6.2, is attempting to communicate with an independent LU 6.2.

Programmer Action

There are thousands of reasons for this return code. A specific reason for the failure can be found in the *SNA sense data* associated with this return code. To diagnose the problem, your program must provide this SNA sense data to its user, or indicate a way it can be obtained.

The local program should try again (at least once) to allocate a session for a conversation. There are some conditions that cause a failure the first time and succeed every time after that. To avoid congesting the network with attempted allocation requests, the local program should pause or wait for a keystroke before repeatedly retrying the allocation, and limit the number of retries. The implication of this return code is that something is wrong at the partner location or with the connection between the two computers, but this is not necessarily always true.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Examine any platform-related message logs and error logs at both locations to find more information about this problem. Look for equipment failures or setup problems related to the network components and the computers using them. Fix the problem, as indicated by any SNA sense data associated with this call. You may need to run the applications again with SNA tracing activated to get the sense data.

After correcting the problem, try running the pair of programs again, if appropriate.

3 CM_CONVERSATION_TYPE_MISMATCH

The conversation type of the local program is not expected or supported by the partner.

Explanation

The partner computer rejected the incoming Attach because it or the partner program does not support the specified conversation type. The local program set the *conversation_type* to CM_MAPPED_CONVERSATION or CM_BASIC_CONVERSATION, and the partner configuration does not support that type of conversation.

This conversation with the partner is over. This return code will recur until either the local program or the partner configuration is changed. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 10086034.

Likely Causes

This is probably a configuration problem at the partner or a mismatched pair of programs. The TP definition at the partner must match the conversation type used by the local program.

Programmer Action

Decide what the correct conversation type should be (basic or mapped). Change the local program so it uses the correct conversation type.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Change the TP definition at the partner to reflect the conversation type required by the local program. To allow any conversation type at the partner, configure the conversation type in the TP definition to allow “Either.” The install and setup procedures for your application should not allow this return code to occur in a production environment.

5 CM_PIP_NOT_SPECIFIED_CORRECTLY

The partner expected PIP data, which cannot be sent using CPI-C.

Explanation

The partner computer rejected the incoming Attach because the partner is defined to expect one or more program initialization parameter (PIP) variables. No PIP data was sent, because it cannot be sent using CPI-C. This return code is returned only when the partner program is using a native APPC (LU 6.2) application programming interface and is not using CPI-C.

This conversation with the partner is over. This return code will recur until either the local program or the partner configuration is changed. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 10086032.

Likely Causes

This is a configuration problem at the partner. The TP definition at the partner cannot require incoming PIP data for this conversation.

Programmer Action

PIP data is an obsolete APPC concept, not supported by CPI-C. The partner program needs to be modified to receive initialization parameters as part of its initial data exchange with its partners.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Change the partner's TP definition so that it does not expect PIP data.

6 CM_SECURITY_NOT_VALID

The conversation security fields sent by the local program are invalid for the partner.

Explanation

The partner computer rejected the user ID or password received on an incoming Attach.

This conversation with the partner is over. This return code will recur until either the values supplied by the local program or the partner configuration are changed. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 080F6051.

Likely Causes

This is probably a configuration problem at the partner or an incorrect user ID and password supplied to the local program. The cause differs depending upon the local *conversation_security_type* of the conversation that failed. There are three values for *conversation_security_type*:

- CM_SECURITY_NONE — the partner's TP definition has been configured to expect conversation security on the incoming Attach, but none was supplied by the local program.
- CM_SECURITY_SAME — this could be an access control problem. For example, the user ID is valid, but the local user does not have the authority to run the partner program.

  If using OS/2 Communications Manager, a likely cause is the use of *Enhanced_Security_Same*. The LAN (locally-logged-on) user ID and password were sent automatically for you. The partner was obligated to check these, and rejected the attempt. The local program should retry the allocation request with CM_SECURITY_NONE.

  Commonly available CPI-C applications like APING, ATELL, and AREXEC default to using *conversation_security_type*(CM_SECURITY_SAME). This return code indicates that the partner does not accept the user ID and password that was sent. You may be able to get the applications to run by starting the local program with *conversation_security_type*(CM_SECURITY_NONE), by using the “-n” command line flag.

  Another possibility occurs when the local program issues an Allocate() call with CM_SECURITY_SAME, but the local platform downgrades the security level to CM_SECURITY_NONE and sends that to the partner. If the partner's TP definition has been configured to require security, this return code results.
- CM_SECURITY_PROGRAM — the combination of user ID and password were not accepted by the partner. For example, the partner is using OS/2 Communications Manager and its TP definition specifies conversation_security(YES). The partner's APPC attach manager searched its list of valid user IDs and passwords, but did not find the received combination.

  User IDs and passwords are both case sensitive. Assure that the values sent by the local program match those expected by the partner's security configuration.

Programmer Action

The Allocate() call made by the local program is using the wrong user_ID or password parameter, or the partner's configuration of allowed user IDs and passwords needs to be changed. In the local program, look at the values specified in the CPI-C Set_Conversation_Security_Type(), Set_Conversation_Security_User_ID(), and Set_Conversation_Security_Password() calls. Verify that they specify user_ID and password values acceptable to the partner.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

If the problem is with the partner configuration, assure that the user ID and password combination sent by the local program has been correctly defined there. User IDs and passwords are case-sensitive. Be sure that the combination of uppercase and lowercase letters configured at the partner matches those specified by the local program.

Assure that if these are required (see the partner's TP definition), they will be accepted by the partner LU (see the partner's definition of its partner LU).

7 CM_SYNC_LVL_NOT_SUPPORTED_LU

The partner's CPI-C platform does not support the sync_level used by the local program.

Explanation

The local computer rejected the Allocate() call (CMALLC) because the local program specified a *sync_level* of CM_SYNC_POINT, which the partner does not support. This return code is returned only for conversations with *sync_level* set to CM_SYNC_POINT (or set to CM_CONFIRM, if activating a session with IMS using its LU 6.1 adapter).

This conversation with the partner is over. This return code will recur until either the local program or the partner configuration is changed. CPI-C reports this return code on an Allocate() call.

Likely Causes

The partner, as it is currently installed, does not support Resource Recovery (sync point) operations. The local program, which specifies sync point operation, may be incompatible with the current system software.

Programmer Action

Since sync point is not available at the partner, consider designing your program logic to do the checkpointing it needs without using the CPI Resource Recovery (CPI-RR) functions.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Upgrade the partner's system software, if possible, to a version that supports sync point operations.

8 CM_SYNC_LVL_NOT_SUPPORTED_PGM

The sync level of the local program is not expected by the partner.

Explanation

The partner computer rejected the incoming Attach because the local program specified a synchronization level (with the *sync_level* parameter) that the partner program does not support.

This conversation with the partner is over. This return code will recur until either the local program or the partner configuration is changed. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 10086041.

Likely Causes

This is probably a configuration problem at the partner or a mismatched pair of programs. For example, the local transaction program issued an Allocate() call with *sync_level* set to CM_CONFIRM, but at the partner computer, the TP definition was configured as sync_level(NONE).

Programmer Action

Decide what the correct *sync_level* should be for the conversation. Change the local program, if necessary, so it uses the correct *sync_level*. The partner's TP definition should match the *sync_level* chosen by the local program.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Change the TP definition at the partner to reflect the sync_level required by the local program.

9 CM_TPN_NOT_RECOGNIZED

The partner computer does not recognize the *TP_name* sent by the local program.

Explanation

The partner computer rejected the incoming Attach because the local program specified a *TP_name* that the partner LU does not recognize.

This return code can also indicate that the partner LU recognized the *TP_name*, but could not start the program for some reason. This can be caused by authorization problems. Some APPC products (like VM/ESA) check that three things match up: LU name, mode name, and TP name. These platforms check the incoming user_ID and password of each defined TP, and reject the incoming Attach with this return code if the Attach is not authorized to start a corresponding program on the partner computer.

This conversation with the partner is over. This return code will recur until either the local program or the partner configuration is changed. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 10086021.

Likely Causes

This is probably a configuration problem at the partner or an incorrect *TP_name* supplied to the local program. The *TP_name* parameter in the partner's TP definition is one of the APPC configuration fields that is case-sensitive. Be sure that the combination of uppercase and lowercase letters matches those specified in the program.

For example:

- Examine the TP definition at the partner. Its TP definition may have the *TP_name* spelled wrong or be using the wrong combination of uppercase and lowercase characters. The *TP_name* configured at the partner must exactly match the *TP_name* used by the local program. (The *TP_name* sent by the local program may be set up in the local side information file.)
- If the partner is using OS/2 Communications Manager, there may be no TP definition that corresponds to the incoming Attach. The attach manager on the partner computer will still attempt to start the program, using the parameters on the DEFINE_DEFAULTS statement (if present). Assure that the correct directory_for_inbound_attaches is specified, along with the correct default_tp_operation and default_tp_program_type.
- If the partner is using the VM operating system, this return code will be received if the remote VM AVS is not authorized via the “*IDENT” control statements in VM to send information to the TP Program. This return code is saying that there is no way to get to the TP Program on VM from OS/2.
- If the partner is using the VM operating system, this return code is returned if an attempt was made to run the partner program in a user's virtual machine (using a non-dedicated private gateway), and the user ID and password were not specified correctly. User ID and password are case-sensitive fields; be sure they are specified in uppercase if that is what the partner expects.

Programmer Action

The *TP_name* sent to the partner is either a field configured in the local CPI-C side information, or a parameter on a Set_TP_Name() call issued by the local program.

- Ask the program user if the correct *symbolic_destination_name* was supplied, if the *TP_name* was configured as part of the local CPI-C side information.
- Assure that your program constructs the *TP_name* and *TP_name_length* parameter correctly, if it issues a Set_TP_Name() call. The characters in the *TP_name* are case sensitive.

Check the designated partner LU and mode names, as well, if required by the partner.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

At the partner, check the list of TP names to be recognized. Assure that they match the values supplied for the *TP_name* values on the Allocate() call in the local computer.

If this checks out (that is, you find the TP correctly defined at the partner), make sure that partner TP is correctly authorized for the user_ID and password sent on the Attach.

10 CM_TP_NOT_AVAILABLE_NO_RETRY

The partner cannot start the program that corresponds to the *TP_name* that was sent.

Explanation

The partner computer rejected the incoming Attach because it could not start the program that corresponds to the *TP_name* it received. Without correcting the partner's configuration or operating environment, retrying the Allocate() call will not succeed.

This conversation with the partner is over. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 084C0000.

Likely Causes

This is probably a configuration problem at the partner. There is a valid TP definition at the partner for the *TP_name* that was sent, but the executable program named in that TP definition could not be started. For example:

- The intended executable program does not exist at the partner location.
- The executable program exists at the partner, but not in the drive, path, or directory specified in the partner's TP definition.
- The executable program exists at the partner, but its name is spelled wrong in the partner's TP definition.
- The operating system, CPI-C, or APPC software at the partner has exhausted its available resources (such as memory, processes, queues, or threads). The partner is unable to start the program that corresponds with the incoming *TP_name*.
- If the partner is using OS/2 Communications Manager, the APPC attach manager at the partner location may have failed to start the program due to an OS/2 error on its internal DosExecPgm() or DosStartSession() function call. Any of the non-zero OS/2 return codes from these function calls may be causing this situation. For example, the partner location is using OS/2 Communications Manager and its TP definition indicates this program is to be attach-manager-started. If there is insufficient memory for the attach manager to start the program, the incoming Attach will be rejected with this return code.

This could similarly occur if the partner's operating system finds that it has consumed all of its processes or screen groups.

Programmer Action

None. Until the partner configuration or operating environment is corrected, it is unlikely that this conversation can be completed.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Assure that the intended program has been defined correctly at the partner (via the TP definition), and that it has been compiled and linked correctly for its operating system. If the partner is logging its attach manager errors, look at its error log for more information.

11 CM_TP_NOT_AVAILABLE_RETRY

The partner cannot now start the program that corresponds to the *TP_name* that was sent.

Explanation

The partner computer rejected the incoming Attach because it could not start the program that corresponds to the *TP_name* it received. Because of timing conditions, retrying the conversation may succeed in causing the partner program to be started.

This conversation with the partner is over. CPI-C reports this return code on a conversation call issued after an Allocate() call.

(An Attach is the internal message that is created when an Allocate() call is successful. It contains the TP name to be sent to the partner, as well as the conversation type and sync level. It also contains any conversation security information.)

You may see this return code associated with SNA sense data 084B6031.

Likely Causes

This is probably a problem with the operating environment or configuration at the partner. There is a valid TP definition at the partner for the *TP_name* that was sent, but the executable program named in that TP definition could not be started. For example:

- The APPC attach manager associated with the partner LU is stopped. In OS/2 Communications Manager, this can be caused by a missing START_ATTACH_MANAGER statement in the Communications Manager (.NDF) node definitions file.
- The incoming_allocate_timeout defined in the TP definition at the partner LU is 0 or is too short. This only applies if the partner TP is configured as Queued.
- The incoming_allocate_queue_depth configured in the TP definition at the partner LU is too small. This only applies if the partner TP is configured as Queued.

- At the partner, the incoming Attach was queued, waiting to be processed. The intended program was remotely started, but ended before it could process the queued incoming Attach. For example, it had not yet issued an Accept_Conversation() call (CMACCP).

Programmer Action

Retry the Attach by reissuing the Initialize_Conversation() call (CMINIT) and Allocate() call (CMALLC). However, to avoid congesting the network with attempted allocation requests, the local program should pause or wait for a keystroke before retrying the conversation.

Check the Accept_Conversation() calls (CMACCP) made by the partner program. It may be failing to make these calls correctly.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Start the APPC attach manager, increase the incoming_allocate_timeout, increase the incoming_allocate_queue_depth in the configuration at the partner, if appropriate.

17 CM_DEALLOCATED_ABEND

The partner has abruptly ended the conversation.

Explanation

This return code is returned under one of the following conditions:

- The partner program issued a Deallocate() call (CMDEAL) with *deallocate_type* set to CM_DEALLOCATE_ABEND, or the partner has done so because of a remote program abnormal-ending condition. If the conversation at the partner program was in **Receive** state when the call was issued, information sent by the local program and not yet received by the partner program is purged.
- The partner program terminated normally but did not deallocate the session before terminating. The CPI-C or APPC software used by the partner deallocated the conversation on behalf of the partner program.

This conversation with the partner is over. This return code is reported to the local program on a call the program issues for a conversation in **Send** or **Receive** state.

You may see this return code associated with SNA sense data 08640000.

Likely Causes

The partner program encountered a condition that caused it to terminate unexpectedly, or it was stopped unexpectedly by a user. For example:

- The partner program was running on OS/2. A user terminated the partner program using Ctrl+Break or Ctrl+C, or any of the menu options for ending programs. OS/2's ExitList processing causes the Deallocate() call to be issued as part of cleaning up the process.
- The partner program was processing an error, and encountered another error. Rather than loop in its error-handling code, the partner program issued a Deallocate() call with *deallocate_type* set to CM_DEALLOCATE_ABEND.

Programmer Action

None.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Correct the problem encountered by the partner program.

18 CM_DEALLOCATED_NORMAL

The partner program has ended the conversation.

Explanation

The partner program issued a Deallocate() call (CMDEAL) with *deallocate_type* set to CM_DEALLOCATE_SYNC_LEVEL or CM_DEALLOCATE_FLUSH. If *deallocate_type* is CM_DEALLOCATE_SYNC_LEVEL, the *sync_level* is CM_NONE.

This conversation with the partner is over. This return code is reported to the local program on a call the program issues for a conversation in **Receive** state.

Likely Causes

This return code generally indicates the successful completion of a conversation.

Programmer Action

None.

If your program sees this return code on a Receive() call (CMRCV), it should examine the value of the *data_received* parameter. Although the conversation is over, data that the partner sent, but which not yet received, may have been returned on the Receive() call.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

None.

19 CM_PARAMETER_ERROR

The local program tried to allocate a session, using invalid parameter names.

Explanation

The local program issued an Allocate() call (CMALLC) that references a parameter containing an invalid conversation characteristic. The source of the argument can be outside the scope of the local program, such as a field in the CPI-C side information that is referenced by the Initialize_Conversation() call.

The state of the conversation remains unchanged.

Note: Contrast this return code with the CM_PROGRAM_PARAMETER_CHECK return code, which indicates a syntax error in a parameter.

Likely Causes

This return code indicates a parameter value that is not valid on the local platform (for example, it has not been configured), or your program doesn't have the correct permission level to use that specific parameter value. This is usually a problem with the partner_LU_name, mode_name, or TP_name being used. It means that one or more fields in the local CPI-C side information are wrong, or a CPI-C Set call—issued after the Initialize_Conversation(), but before the Allocate() call—contained an invalid name. For example,

- The local program specified an all-blank symbolic destination name on its Initialize_Conversation() call, but the program failed to issue Set calls for the partner LU name, mode name, and TP name.
- The partner_LU_name, mode_name, or TP_name contain one or more characters that are not valid for these names on the local computer. For example, a tilde (˜) is not a valid character in these names.
- The local program attempted to use one of APPC's special modes, CPSVCMG CPSVRMGR, or SNASVCMG, which are not for use by application programs.
- The local program is using OS/2 Communications Manager. If you specified a partner_LU_name without a period, in the side information or on a Set_Partner_LU_Name() call, the Communications Manager interprets that name as a *partner_LU_alias*. If that *partner_LU_alias* is not configured on the local computer at the time of the Allocate() call, this return code results.

Programmer Action

If the local program is using CPI-C side information via a symbolic destination name, consider overriding the partner LU name or mode name values. If the partner LU name or mode name was set using the Set_Partner_LU_Name() or Set_Mode_Name() calls, consider requesting different names from the user.

You may want to consider checking the validity of the characters passed to CPI-C in the Set_Partner_LU_Name(), Set_Mode_Name(), and Set_TP_Name() calls, since CPI-C does no validity checking of these calls.

Receiving the return code does not invalidate the conversation ID. If you can set different names, given input from a user, you can retry the Allocate() call without issuing Initialize_Conversation() again.

Operator Action

Assure that the partner LU name, mode name, and other conversation characteristics in the local CPI-C side information are set up properly, and correspond to the local APPC configuration.

20 CM_PRODUCT_SPECIFIC_ERROR

The local program made a CPI-C call that failed because of a platform-related condition.

Explanation

An operational problem has been detected which relates to the CPI-C product that the local program is using. A description of the error has been written to the local CPI-C product's system error log. See the product documentation for an indication of conditions and state changes caused by this return code.

The state of the conversation remains unchanged.

Likely Causes

This return code generally indicates a problem with the underlying CPI-C software and the environment in which it is running, not the local program. For example:

- The underlying CPI-C or APPC software is stopped, has not been started or is not yet fully active, or is abended.
- CPI-C is unable to allocate needed memory for its internal control blocks. For example, no more conversation IDs are available.
- If the return code is from a call to Accept_Conversation() (CMACCP), it is likely that the program was started locally (for example, from the command line), but the program should have been started by the attach manager because of an incoming Attach.
- There is a CPI-C implementation bug.

Here are some other platform-specific examples of this return code.

On the OS/2 Communications Manager:

If the local platform is OS/2 Communications Manager, the local TP may have been deactivated using Subsystem Management.

If the local platform is OS/2 Communications Manager, the `APPCTPN` environment variable was not set before calling Accept_Conversation().

On CICS/ESA:

If the local platform is CICS/ESA, this return code results in one of the following informational error messages:

- DFHCP0742 - the session is not available for CPI-C, as it is already in use by another process.
- DFHCP0743 - CPI-C can not be used, as the transaction was initiated by ATI.
- DFHCP0750 - an unrecognized profile name was supplied in the partner resource *sym_dest_name*.

On MVS/ESA:

If the local platform is MVS/ESA, the system writes a symptom record to SYS1.LOGREC to identify the error.

On VM/ESA:

If the local platform is VM/ESA, the system writes a symptom record to CPICOMM LOGDATA to identify the error.

However, depending on the platform, there are some types of problems that are caused by an **application program**. For example:

- There is a multi-tasking interference problem. One way this can occur is for a multi-threaded program to issue more than one CPI-C call for a given conversation at the same time.

Programmer Action

This return code generally indicates a problem that the local program cannot recover from without intervention. Since the state of the conversation generally is not changed by this call, the local program should issue a Deallocate() call (with *deallocate_type* set to CM_DEALLOCATE_ABEND, if possible) to end the conversation.

Operator Action

First, make sure the local CPI-C software is active. If the CPI-C software was active when this return code occurred, check the local CPI-C product's system error log (and related message logs) for additional details about this problem.

CPI-C may be making calls to underlying APPC software. The details of these calls, including their control blocks and internal return codes, provide lots of diagnostic information. If possible, get a trace of these internal calls, to see the return codes returned by APPC.

21 CM_PROGRAM_ERROR_NO_TRUNC

The partner program issued a Send_Error() call while it was in **Send** state.

Explanation

The partner program issued a Send_Error() call (CMSERR). The partner conversation was in **Send** state; the Send_Error() that it issued did not truncate a logical record. (No truncation occurs when a program issues a Send_Error() call before sending any logical records or after sending a complete logical record.) The local conversation is now in **Receive** state.

You may see this return code associated with SNA sense data 08890000.

Likely Causes

The partner program issued a Send_Error() call. The partner program encountered a problem while it had the permission to send. This could be a problem with the data it was sending or building, or this could be a problem discovered by the partner program's logic.

Programmer Action

The local program should issue a Receive() call to receive a logical record containing a description of the error from its partner. (This presumes that the partner program follows the recommended practice of sending a description of the error it has just encountered after issuing a Send_Error() call.)

The partner program may have been designed to call Send_Error() when it finds a bug in its own logic. Examine carefully why the partner issued Send_Error().

Operator Action

The partner program generally issues a Send_Error() call when it has a problem with its logic or the data it is handling. It may be necessary to check the validity of the data being processed by the partner.

22 CM_PROGRAM_ERROR_PURGING

The partner program issued a Send_Error() call while the partner was in **Receive** or **Confirm** state.

Explanation

One of the following occurred:

- The partner program issued a Send_Error() call (CMSERR) while it was in **Receive** or **Confirm** state. The call may have caused information enroute to the partner program to be purged (discarded), but not necessarily.

  Purging occurs when the partner program issues a Send_Error() call for a conversation in **Receive** state before receiving all the information being sent by the local program. No purging occurs when the partner program issues a Send_Error() call for a conversation in **Receive** state if the partner program has already received all the information sent by the local program. Also, no purging occurs when the partner program issues Send_Error() for a conversation in **Confirm** state.

  When information is purged, the purging can occur at the local system, the partner system, or both.

- The partner program issued a Send_Error() call. The conversation for the partner program was in **Send-Pending** state. No purging of data has occurred. This return code indicates that the partner program had the *error_direction* conversation characteristic set to CM_RECEIVE_ERROR when the Send_Error() call was made.

This return code is normally reported to the local program on a call the program issues after sending some information to the partner program. However, the return code can be reported on a call the program issues before sending any information, depending on the call and when it is issued. The local conversation is now in **Receive** state.

You may see this return code associated with SNA sense data 08890000.

Likely Causes

The partner program issued a Send_Error() call. The partner program encountered a problem while the local program had the permission to send. This could be a problem with the data the partner was receiving or parsing, or this could be a problem discovered by the partner program's logic.

This can also indicate a “race” condition, where both programs issued a Send_Error() call at the same time. The local program lost the race.

Programmer Action

The local program should issue a Receive() call, to receive a logical record containing a description of the error from its partner. (This presumes that the partner program follows the recommended practice of sending a description of the error it has just encountered after issuing a Send_Error() call.)

The partner program may have been designed to call Send_Error() when it finds a bug in its own logic. Examine carefully why the partner issued Send_Error().

Operator Action

The partner program generally issues a Send_Error() call when it has a problem with its logic or the data it is handling. It may be necessary to check the validity of the data being processed by the partner.

23 CM_PROGRAM_ERROR_TRUNC

The partner program issued a Send_Error() call for a basic conversation, truncating a logical record it was sending.

Explanation

The partner program issued a Send_Error() call (CMSERR) while in **Send** state, and the Send_Error() truncated a logical record. Truncation occurs when a program begins sending a logical record and then issues a Send_Error() call before sending the complete logical record.

CPI-C reports this return code to the local program on a Receive() call (CMRCV), after receiving the initial portion of the truncated logical record. The local conversation remains in **Receive** state.

This return code is returned for basic conversations only.

You may see this return code associated with SNA sense data 08890001.

Likely Causes

The partner program issued a Send_Error() call. The partner program was in the middle of sending a logical record, and it encountered a problem. This could be a problem with the data or a problem discovered by the partner program's logic.

Programmer Action

The local program should issue a Receive() call to receive a logical record containing a description of the error from its partner. (This presumes that the partner program follows the recommended practice of sending a description of the error it has just encountered after issuing a Send_Error() call.)

The partner program may have been designed to call Send_Error() when it finds a bug in its own logic. Examine carefully why the partner issued Send_Error().

Operator Action

The partner program generally issues a Send_Error() call when it has a problem with its logic or the data it is handling. It may be necessary to check the validity of the data being processed by the partner.

24 CM_PROGRAM_PARAMETER_CHECK

The local program called CPI-C with an invalid parameter.

Explanation

The local program issued a CPI-C call with an error in one or more parameters. (“Parameters” include not only the parameters described as part of the call syntax, but also the CPI-C conversation characteristics associated with the *conversation_ID*.)

If the return code is from a call to Initialize_Conversation() (CMINIT), it is likely that the symbolic destination name supplied by the user was spelled wrong, or the name has not been configured in the local CPI-C side information.

The source of the error is considered to be under the control of the local program. This return code may be caused by the failure of the program to pass a valid parameter address. The program should not examine any other returned variables associated with the call.

The state of the conversation remains unchanged.

Likely Causes

This can be caused by any incorrect parameter. All parameters passed to CPI-C must be valid pointers to variable fields of the proper length. Other examples of parameter errors are:

- The symbolic destination name supplied on the Initialize_Conversation() call was not found in the local CPI-C side information.

  Either the supplied symbolic destination name was spelled wrong (or the wrong mix of upper- and lowercase letters was used), or the supplied symbolic destination name has not been set up in the local CPI-C side information.
- The *conversation_ID* is invalid. This can easily occur if a program issues any CPI-C call after a conversation is over, using the old *conversation ID*.
- The length of a buffer to be sent or received is too large.
- A length field is too large or is out of range.
- An enumerated value is out of range.
- The requested *sync_level* is not supported by the local machine.

Programmer Action

If your program gets its symbolic destination name from a user or an input file, it is likely that this return code is because of an error related to that name. Assure the proper combination of upper and lower case letters, numbers, and symbols (many platforms allow only uppercase letters and numbers). Also, have the user check to see that the symbolic destination name they provided is actually configured on the local computer in the CPI-C side information.

For any call other than Initialize_Conversation(), this return code indicates a bug or logic defect in the local program. Design your program so that users only see this return code when there is a bug to be reported and fixed. For example, do validity and range checking on length values before making CPI-C calls to diagnose potential problems,

Find the CPI-C call that failed, and carefully check the syntax of each parameter that is being supplied. Assure that all parameters are valid pointers to valid fields, and that each of these fields has the proper size and value.

Operator Action

Check that the symbolic destination name that was used is configured in the local CPI-C side information. Also, assure that the symbolic destination supplied to the local program was spelled correctly, and used the correct combination of upper and lower case letters and numbers.

Otherwise, report this as a bug to the supplier of the program.

25 CM_PROGRAM_STATE_CHECK

The local program called CPI-C with the wrong call at this time.

Explanation

This return code is returned under one of the following conditions:

- The local program issued a CPI-C call in a conversation state that was not valid for that call.
- The local program issued a Wait_For_Conversation() call, and there were no outstanding operations for any conversations known to the local program.
- For a conversation with *sync_level* set to CM_SYNC_POINT, the conversation's context is in the **Backout-Required** condition. The call issued is not allowed for this conversation while its context is in this condition.

The state of the conversation remains unchanged.

Likely Causes

For production-level (debugged) programs, this is usually caused when the wrong two programs are talking to one another. At the initiating side, this may be caused because the wrong symbolic destination name is specified by the initiating program. Similarly, the partner LU name or the TP name may be wrong. At the target side, TP definition may specify the wrong program to be started.

If your program has issued the Wait_For_Conversation() call, this return code should be handled as part of the mainline program logic. There were no conversations with outstanding calls on which to wait for completion.

Otherwise, this return code indicates a bug or logic defect in the local program. This generally occurs when CPI-C calls are issued in the wrong order, or when interactions with the partner program have been overlooked. For example:

- The Set_Mode_Name() call is only valid in **Initialize** state, that is, after issuing Initialize_Conversation() but before issuing Allocate(). If it is issued after Allocate() or Accept_Conversation(), CPI-C returns this return code.
- After receiving a return code that indicates the partner program has issued a Send_Error() call (for example, CM_PROGRAM_ERROR_NO_TRUNC or CM_PROGRAM_ERROR_PURGING), the local program is in **Receive** state. It can only issue CPI-C calls that are valid in **Receive** state.

On the OS/2 Communications Manager

This return code is also returned on an Accept_Conversation() call for the following reasons:

- The operator or program set a TP name in the `APPCTPN` environment variable that was incorrect, that is, it did not match the TP name on the incoming Attach.
- An operator-started program issued an Accept_Conversation() call, but the call expired before the incoming Attach arrived. The duration that a call waits for an incoming Attach is configured on the TP definition, using the *receive_allocate_timeout* field.

Programmer Action

This is readily caused when the user of the initiating program specifies a valid, but wrong, symbolic destination name, which caused this program to be erroneously started. Convey the probable program mismatch to the program's user, and end the program using a Deallocate() call with *deallocate_type* set to CM_DEALLOCATE_ABEND.

If your program sees this on a Wait_For_Conversation() call, it should handle it as an expected return code. It means that there were no outstanding operations on any conversation. This can easily occur when all the conversations are using blocking mode, that is, their *processing_mode* conversation characteristic is set to the default value of CM_BLOCKING.

Otherwise, users of your program should never see this return code; to users, this return code indicates a bug. The program should report to its user that a bug has been encountered, and log the bug. The program should not examine any other returned variables associated with the call.

The conversation ID for this conversation is still valid. The local program can find the current state of the conversation by issuing the CPI-C Extract_Conversation_State() call. Refer to a CPI-C reference manual for a list of calls that are allowed in each conversation state.

Operator Action

Assure that the TP definition at the target computer specifies the correct program to be started for the arriving TP name. Also, assure that the program on the initiating side has obtained the correct symbolic destination name, and that the side information for that entry is correct.

Otherwise, report this as a bug to the supplier of the local program.

26 CM_RESOURCE_FAILURE_NO_RETRY

The active conversation has been unexpectedly ended, and starting it again probably will fail.

Explanation

A permanent failure prematurely ended the conversation. The condition is not temporary; operator intervention is required to correct the problem.

This conversation with the partner is over. This return code can be reported to the local program on a call it issues for a conversation in any state other than **Reset** or **Initialize**.

Likely Causes

The session or link used to get to the partner has been broken, or the partner program was abruptly stopped. For example:

- The local and partner LUs have been disconnected from each other. The link or session has been abruptly terminated.
- The partner program was deactivated while it was running. For example, if the partner is using OS/2 Communications Manager, this can be done using Communications Manager Subsystem Management.
- The partner computer may have violated internal SNA protocols.

Programmer Action

Without human intervention, the conversation probably cannot be successfully restarted.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Examine any platform-related message logs and error logs at both locations to find more information about this problem. Look for equipment failures or setup problems related to the network components and the computers using them. Fix the problem, as indicated by any SNA sense data associated with this call. You may need to run the applications again with SNA tracing activated to get the sense data.

After correcting the problem, try running the pair of programs again, if appropriate.

27 CM_RESOURCE_FAILURE_RETRY

The active conversation has been unexpectedly ended, and starting it again may be successful.

Explanation

A temporary failure prematurely ended the conversation.

This conversation with the partner is over. This return code can be reported to the local program on a call it issues for a conversation in any state other than **Reset** or **Initialize**.

Likely Causes

Some aspect of the partner computer, needed for CPI-C communication, has been deactivated. Here are some examples of how this return code is caused.

- The partner computer was powered off or re-booted during the conversation.
- The partner computer stopped or unloaded its APPC software during the conversation.
- The partner computer deactivated its data link control (DLC) during the conversation.
- The local or partner LU deactivated the session in the middle of a conversation. In OS/2, for example, this can be done using Communications Manager Subsystem Management or issuing the CNOS verb.
- The local LU was notified of a session outage occurring in the network.

Programmer Action

The local program should attempt to allocate a session again. It is possible for some permanent failures to be initially reported as temporary, but the allocation of subsequent conversations would also fail.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Examine any platform-related message logs and error logs at both locations to find more information about this problem. Look for equipment failures or setup problems related to the network components and the computers using them. Fix the problem, as indicated by any SNA sense data associated with this call. You may need to run the applications again with SNA tracing activated to get the sense data.

After correcting the problem, try running the pair of programs again, if appropriate.

28 CM_UNSUCCESSFUL

The local program made a CPI-C call that could not complete its requested function.

Explanation

There are three CPI-C calls that can fail to complete their desired function when they are issued, but may succeed if issued later. For these three calls, this return code should be handled as part of the mainline program logic.

The state of the conversation remains unchanged.

Likely Causes

This return code has different meanings, depending on whether the CPI-C call that was issued was Allocate(), Receive(), or Test_Request_To_Send().

- For an Allocate() call (CMALLC) with return_control(CM_IMMEDIATE), CPI-C could not obtain a contention-winner session to the partner computer, using the requested mode name.
- For a Receive() call (CMRCV) with *receive_type* set to CM_RECEIVE_IMMEDIATE, there is no data to receive, and the status_received value hasn't changed since the last Receive().
- For a Test_Request_To_Send() call (CMTRTS), no request-to-send signal has been received from the partner since the last time the program checked for the signal.

Programmer Action

The requested call was unable to complete the function that was requested, but no explicit action is required. Retry the call as appropriate. If you are issuing these calls in a programmed loop, consider pausing slightly within the loop, to avoid hard CPU usage (thus allowing other programs a chance to run).

Operator Action

If this return code is returned on an Allocate() call, it may indicate that there are a large number of simultaneous sessions between this computer and its partner using the same mode. These sessions might be spread among many programs that are running at the same time. Consider changing the mode definition on both computers for the mode name in question, so it has more contention-winner sessions available concurrently.

30 CM_DEALLOCATED_ABEND_SVC

The partner program was abnormally terminated, or it ended the conversation abnormally by issuing a Deallocate() call.

Explanation

This return code is returned under one of the following conditions:

- The partner program, using a native APPC (LU 6.2) application programming interface and not using CPI-C, issued a DEALLOCATE verb specifying a TYPE parameter of ABEND_SVC. If the conversation for the partner program was in

Receive state when the verb was issued, information sent by the local program and not yet received by the partner program is purged.

- The partner program either terminated abnormally or terminated normally but did not deallocate the conversation before terminating. The CPI-C or APPC software used by the partner deallocated the conversation on behalf of the partner program.

This return code is returned for basic conversations only.

This conversation with the partner is over. This return code is reported to the local program on a call the program issues for a conversation in **Send** or **Receive** state.

You may see this return code associated with SNA sense data 08640000.

Likely Causes

Here are some examples of how this return code is caused.

- If the partner program is running on OS/2, this return code is seen locally if the partner program ended unexpectedly because of a protection fault.
- If the partner program is running on the AS/400, this return code reports the absence of a valid routing entry in a subsystem. If no specific subsystem device or remote location name has been entered, check to see if the subsystem QCMN is active. If QCMN is active, be sure the routing entry for APPC programs (PGMEVOKE) is present in the subsystem. If a subsystem other than QCMN is being used and a specific communications entry or remote location name entry is being used, be sure a routing entry with PGMEVOKE is specified in the same subsystem. In both cases, the fix is to add a routing entry in a subsystem.

Programmer Action

None.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

None.

31 CM_DEALLOCATED_ABEND_TIMER

The partner program ended the conversation abnormally by issuing a Deallocate() call.

Explanation

The remote APPC transaction program issued a DEALLOCATE verb specifying a TYPE parameter of ABEND_TIMER. If the local conversation was in **Receive** state when the verb was issued, information sent by the local program and not yet received by the remote program is purged.

This return code is returned for basic conversations only. In addition, it is returned only when the remote transaction program is using a native APPC (LU 6.2) application programming interface and is not using CPI-C.

This conversation with the partner is over. This return code is reported to the local program on a call the program issues for a conversation in **Send** or **Receive** state.

Likely Causes

The partner program encountered a condition that caused it to choose to terminate abnormally.

Programmer Action

None.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

None.

32 CM_SVC_ERROR_NO_TRUNC

The partner program issued a Send_Error() call while it was in **Send** state.

Explanation

The partner program issued a Send_Error() call with a TYPE parameter of SVC. The partner conversation was in **Send** state; the Send_Error() that it issued did not truncate a logical record. (No truncation occurs when a program issues a Send_Error() call before sending any logical records or after sending a complete logical record.) The local conversation is now in **Receive** state.

This return code is returned for basic conversations only. In addition, it is returned only when the remote transaction program is using a native APPC (LU 6.2) application programming interface and is not using CPI-C.

You may see this return code associated with SNA sense data 08890100.

Likely Causes

The partner program encountered a problem while it had the permission to send. This could be a problem with the data it was sending or building, or this could be a problem discovered by the partner program's logic.

Programmer Action

The local program should issue a Receive() call to receive a logical record containing a description of the error from its partner. (This presumes that the partner program follows the recommended practice of sending a description of the error it has just encountered after issuing a Send_Error() call.)

The partner program may have been designed to call Send_Error() when it finds a bug in its own logic. Examine carefully why the partner issued Send_Error().

Operator Action

The partner program generally issues a Send_Error() call when it has a problem with its logic or the data it is handling. It may be necessary to check the validity of the data being processed by the partner.

33 CM_SVC_ERROR_PURGING

The partner program issued a Send_Error() call while it was in **Receive** or **Confirm** state.

Explanation

One of the following occurred:

- The partner program issued a Send_Error() call with a TYPE parameter of SVC. The conversation for the partner program was in **Receive** or **Confirm** state. The call may have caused information enroute to the partner program to be purged (discarded), but not necessarily.

  Purging occurs when the partner program issues a Send_Error() call for a conversation in **Receive** state before receiving all the information being sent by the local program. No purging occurs when the partner program issues a Send_Error() call for a conversation in **Receive** state if the partner program has already received all the information sent by the local program. Also, no purging occurs when the partner program issues Send_Error() for a conversation in **Confirm** state.

  When information is purged, the purging can occur at the local system, the partner system, or both.

- The partner program issued a Send_Error() call with a TYPE parameter of SVC. The conversation for the partner program was in **Send-Pending** state. No purging of data has occurred. This return code indicates that the partner program had an *error_direction* characteristic set to CM_RECEIVE_ERROR when the Send_Error() call was made.

This return code is normally reported to the local program on a call the program issues after sending some information to the partner program. However, the return code can be reported on a call the program issues before sending any information, depending on the call and when it is issued. The local conversation is now in **Receive** state.

This return code is returned for basic conversations only. In addition, it is returned only when the remote transaction program is using a native APPC (LU 6.2) application programming interface and is not using CPI-C.

You may see this return code associated with SNA sense data 08890100.

Likely Causes

The partner program encountered a problem while the local program had the permission to send. This could be a problem with the data it was receiving or parsing, or this could be a problem discovered by the partner program's logic.

Programmer Action

The local program should issue a Receive() call, to receive a logical record containing a description of the error from its partner. (This presumes that the partner program follows the recommended practice of sending a description of the error it has just encountered after issuing a Send_Error() call.)

The partner program may have been designed to call Send_Error() when it finds a bug in its own logic. Examine carefully why the partner issued Send_Error().

Operator Action

The partner program generally issues a Send_Error() call when it has a problem with its logic or the data it is handling. It may be necessary to check the validity of the data being processed by the partner.

34 CM_SVC_ERROR_TRUNC

The partner program issued a Send_Error() call for a basic conversation, truncating a logical record it was sending.

Explanation

The partner program issued a Send_Error() call with a TYPE parameter of SVC. The conversation for the partner program was in **Send** state. The Send_Error() truncated a logical record. Truncation occurs when a program begins sending a logical record and then issues a Send_Error() call before sending the complete logical record.

CPI-C reports this return code to the local program on a Receive() call, after receiving the initial portion of the truncated logical record. The local conversation remains in **Receive** state.

This return code is returned for basic conversations only. In addition, it is returned only when the remote transaction program is using a native APPC (LU 6.2) application programming interface and is not using CPI-C.

You may see this return code associated with SNA sense data 08890101.

Likely Causes

The partner program was in the middle of sending a logical record, and it encountered a problem. This could be a problem with the data or a problem discovered by the partner program's logic.

Programmer Action

The local program should issue a Receive() call to receive a logical record containing a description of the error from its partner. (This presumes that the partner program follows the recommended practice of sending a description of the error it has just encountered after issuing a Send_Error() call.)

The partner program may have been designed to call Send_Error() when it finds a bug in its own logic. Examine carefully why the partner issued Send_Error().

Operator Action

The partner program generally issues a Send_Error() call when it has a problem with its logic or the data it is handling. It may be necessary to check the validity of the data being processed by the partner.

35 CM_OPERATION_INCOMPLETE

The local program issued a CPI-C call that has not yet completed.

Explanation

A non-blocking operation has been started on the conversation but is not complete. This return code is returned when *processing_mode* is set to CM_NON_BLOCKING for the conversation and the call is suspended waiting for incoming data, buffers, or other resources. A program must use the Wait_For_Conversation() call to wait for the operation to complete and to retrieve the return code for the completed operation.

The state of the conversation remains unchanged.

Likely Causes

This is an “expected” return code. If your program is using non-blocking processing mode, this return code may be returned on any CPI-C call. Your program requested that CPI-C return immediately on each call, even if they have not completed. This is the return code your program sees is the call has not completed. For example:

- Your program issued a Receive() call with the *receive_type* set to CM_RECEIVE_AND_WAIT, and there is nothing yet to receive.
- Your program issued a Send_Data() call and the internal buffers are full and have not yet been sent.

Programmer Action

This return code should be handled as part of the mainline program logic for programs that issue non-blocking calls. The local program was presumably designed to use non-blocking processing mode so that it could do other work rather than waiting on the completion of CPI-C calls. Your program should do whatever work it has available to do, then come back to this conversation and issue a Wait_For_Conversation() call.

Operator Action

None.

36 CM_SYSTEM_EVENT

A platform-specific event has canceled a Wait_For_Conversation() call issued by the local program.

Explanation

A CPI-C Wait_For_Conversation() call was being executed by the local program when an event (such as a signal) occurred. This event was handled by the local program. Wait_For_Conversation() returns this return code to allow the program to reissue the Wait_For_Conversation() call or to perform other processing. It is the responsibility of the event-handling portion of the local program to capture sufficient information for the program to decide how to proceed when it sees this return code.

The state of the conversation remains unchanged.

Likely Causes

The local platform or product has determined that there is an event that your local program should handle. For example, keyboard input must be captured before continuing with a CPI-C Receive() call.

Programmer Action

Handle the interrupt raised by the local platform (in a product-specific manner), then issue a Wait_For_Conversation() call again.

Operator Action

All system-related events that cause this return code should be handled by the local program. When the local program cannot handle or repair an event-related condition, operator intervention may be required.

37 CM_OPERATION_NOT_ACCEPTED

The local program made a CPI-C call while a previously-issued CPI-C call had not yet completed.

Explanation

A previous CPI-C call on this conversation is incomplete. This return code is returned when there is an outstanding operation on the conversation, as indicated by the CM_OPERATION_INCOMPLETE return code to a previous call. On an operating system that supports multiple program threads, when one thread has started an operation that has not completed, this return code may be returned on a call made by another thread on the same conversation.

The state of the conversation remains unchanged.

Likely Causes

The local program issued a CPI-C call when another had not yet completed. This is a bug in the design or coding of the local program.

Programmer Action

This return code indicates a bug or logic defect in the local program. Users of your program should not see this return code; to users, this return code signals a bug to be reported and fixed.

Operator Action

Report this as a bug to the supplier of the local program.

CPI-RR Return Codes (100 and higher)

The following return codes apply only to CPI-C programs that use CPI Resource Recovery (CPI-RR) calls.

For programs using conversations with *sync_level* set to CM_SYNC_POINT, all return codes indicating that backout processing is required have numeric values equal to or greater than CM_TAKE_BACKOUT. This allows you to test for a range of return code values to determine if backout processing is required. An example is:

return_code >= CM_TAKE_BACKOUT

100 CM_TAKE_BACKOUT

A Backout() request has been made; the local program needs to handle it.

Explanation

The remote program, the local system, or the remote system issued a CPI-RR Backout() call, and the local application must issue a Backout() call to restore all protected resources for a context to their status as of the last synchronization point. The conversation's context is in the **Backout-Required** condition upon receipt of this return code. Once the local program issues a Backout() call, the conversation is placed in the state it was in at the time of the last sync point operation.

This return code is returned only for conversations with *sync_level* set to CM_SYNC_POINT.

Likely Causes

The partner program issued a Backout() call.

Programmer Action

Your program should return to the point where its last synchronization point with its partners. The conversation state is now what it was then. To handle what has occurred since that time, all intervening logic and code must be executed again.

Operator Action

None.

130 CM_DEALLOCATED_ABEND_BO

The partner has abruptly ended the conversation.

Explanation

The remote program issued a Deallocate() call (CMDEAL) with *deallocate_type* set to CM_DEALLOCATE_ABEND, or the remote system has done so because of a remote program abnormal-ending condition. If the conversation for the remote program was in **Receive** state when the call was issued, information sent by the local program and not yet received by the remote program is purged.

This conversation with the partner is over.

This return code is returned only for conversations with *sync_level* set to CM_SYNC_POINT.

Likely Causes

The partner program encountered a condition that caused it to terminate unexpectedly, or it was stopped unexpectedly by a user. For example:

- The partner program was processing an error, and encountered another error. Rather than loop in its error-handling code, the partner program issued a Deallocate() call with *deallocate_type* set to CM_DEALLOCATE_ABEND.

Programmer Action

The local conversation's context is in the **Backout-Required** condition and the program must issue a CPI-RR Backout() call to restore all of the context's protected resources to their status as of the last synchronization point.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Correct the problem encountered by the partner program.

131 CM_DEALLOCATED_ABEND_SVC_BO

The partner program was abnormally terminated, or it ended the conversation abnormally by issuing a Deallocate() call.

Explanation

This return code is returned under one of the following conditions:

- The partner program, using a native APPC (LU 6.2) application programming interface and not using CPI-C, issued a DEALLOCATE verb specifying a TYPE parameter of ABEND_SVC. If the conversation for the partner program was in **Receive** state when the verb was issued, information sent by the local program and not yet received by the partner program is purged.
- The partner program either terminated abnormally or terminated normally but did not deallocate the conversation before terminating. The CPI-C or APPC software used by the partner deallocated the conversation on behalf of the partner program.

This return code is returned for basic conversations only. In addition, it is returned only for conversations with *sync_level* set to CM_SYNC_POINT.

This conversation with the partner is over. This return code is reported to the local program on a call the program issues for a conversation in **Send** or **Receive** state.

You may see this return code associated with SNA sense data 08640000.

Likely Causes

Here are some examples of how this return code is caused.

- This return code is seen locally if the partner program ended unexpectedly because of a protection fault or other operating system exception.
- If the partner program is running on the AS/400, this return code reports the absence of a valid routing entry in a subsystem. If no specific subsystem device or remote location name has been entered, check to see if the subsystem QCMN is active. If QCMN is active, be sure the routing entry for APPC programs (PGMEVOKE) is present in the subsystem. If a subsystem other than QCMN is being used and a specific communications entry or remote location name entry is being used, be sure a routing entry with PGMEVOKE is specified in the same subsystem. In both cases, the fix is to add a routing entry in a subsystem.

Programmer Action

The local conversation's context is in the **Backout-Required** condition and the program must issue a CPI-RR Backout() call to restore all of the context's protected resources to their status as of the last synchronization point.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

None.

132 CM_DEALLOCATED_ABEND_TIMER_BO

The partner program ended the conversation abnormally by issuing a Deallocate() call.

Explanation

The remote APPC (LU 6.2) transaction program issued a DEALLOCATE verb specifying a TYPE parameter of ABEND_TIMER. If the conversation for the remote program was in **Receive** state when the verb was issued, information sent by the local program and not yet received by the remote program is purged. This return code is reported to the local program on a call the program issues for a conversation in **Send** or **Receive** state.

This conversation with the partner is over.

This return code is returned for basic conversations only. In addition, it is returned only for conversations with *sync_level* set to CM_SYNC_POINT, and only when the remote transaction program is using a native APPC (LU 6.2) application programming interface and is not using CPI-C.

Likely Causes

The partner program encountered a condition that caused it to choose to terminate abnormally.

Programmer Action

The local conversation's context is in the **Backout-Required** condition and the program must issue a CPI-RR Backout() call to restore all of the context's protected resources to their status as of the last synchronization point.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

None.

133 CM_RESOURCE_FAIL_NO_RETRY_BO

The active conversation has been unexpectedly ended, and starting it again probably will fail.

Explanation

A failure occurred that caused the conversation to be prematurely terminated. For example, the session being used for the conversation was deactivated because of a session protocol error, or the conversation was deallocated because of a protocol error between the mapped conversation components of the systems. The condition is not temporary, and the program should not retry the transaction until the condition is corrected. This return code can be reported to the local program on a call it issues for a conversation in any state other than **Reset** or **Initialize**.

This conversation with the partner is over.

This return code is returned only for conversations with *sync_level* set to CM_SYNC_POINT.

Likely Causes

The session or link used to get to the partner has been broken, or the partner program was abruptly stopped. For example:

- The local and partner LUs have been disconnected from each other. The link or session has been abruptly terminated.
- The partner program was deactivated while it was running.
- The partner computer may have violated internal SNA protocols.

Programmer Action

The local conversation's context is in the **Backout-Required** condition and the program must issue a CPI-RR Backout() call to restore all of the context's protected resources to their status as of the last synchronization point.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Examine any platform-related message logs and error logs at both locations to find more information about this problem. Look for equipment failures or setup problems related to the network components and the computers using them. Fix the problem, as indicated by any SNA sense data associated with this call. You may need to run the applications again with SNA tracing activated to get the sense data.

After correcting the problem, try running the pair of programs again, if appropriate.

134 CM_RESOURCE_FAILURE_RETRY_BO

The active conversation has been unexpectedly ended, and starting it again may be successful.

Explanation

A failure occurred that caused the conversation to be prematurely terminated. For example, the session being used for the conversation was deactivated because of a session outage such as a line failure, a modem failure, or a cryptography failure. The condition may be temporary, and the program can retry the transaction. This return code can be reported to the local program on a call it issues for a conversation in any state other than **Reset** or **Initialize**.

This conversation with the partner is over.

This return code is returned only for conversations with *sync_level* set to CM_SYNC_POINT.

Likely Causes

Some aspect of the partner computer, needed for CPI-C communication, has been deactivated. Here are some examples of how this return code is caused.

- The partner computer was powered off or re-booted during the conversation.
- The partner computer stopped or unloaded its APPC software during the conversation.
- The partner computer deactivated its data link control (DLC) during the conversation.

- The local or partner LU deactivated the session in the middle of a conversation.
- The local LU was notified of a session outage occurring in the network.

Programmer Action

The local program should attempt to allocate a session again. It is possible for some permanent failures to be initially reported as temporary, but the allocation of subsequent conversations would also fail.

The local conversation's context is in the **Backout-Required** condition and the program must issue a CPI-RR Backout() call to restore all of the context's protected resources to their status as of the last synchronization point.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action

Examine any platform-related message logs and error logs at both locations to find more information about this problem. Look for equipment failures or setup problems related to the network components and the computers using them. Fix the problem, as indicated by any SNA sense data associated with this call. You may need to run the applications again with SNA tracing activated to get the sense data.

After correcting the problem, try running the pair of programs again, if appropriate.

135 CM_DEALLOCATED_NORMAL_BO

The local program issued a Send_Error() call, but the conversation with the partner had already been deallocated.

Explanation

When a Send_Error() call (CMSERR) is issued in **Receive** state, incoming information is purged by the system. This purged information may include a notification of Deallocate-Abend from the remote program or system. When such a notification is purged, CPI-C returns CM_DEALLOCATED_NORMAL_BO instead of one of the following return codes:

- CM_DEALLOCATED_ABEND_BO,
- CM_DEALLOCATED_ABEND_SVC_BO, or
- CM_DEALLOCATED_ABEND_TIMER_BO.

This conversation with the partner is over.

This return code is returned only for conversations with *sync_level* set to CM_SYNC_POINT.

Likely Causes

The partner program deallocated the conversation unexpectedly.

Programmer Action

The local conversation's context is in the **Backout-Required** condition and the program must issue a CPI-RR Backout() call to restore all of the context's protected resources to their status as of the last synchronization point.

The conversation is now in **Reset** state; the *conversation_ID* that was supplied in this CPI-C call is no longer valid.

Operator Action
None.

Names to Numbers, in Alphabetical Order

Sometimes you have a return code's number; other times you have its name. The descriptions above were listed in numerical order; the following list is in alphabetical order, so you can map from a return code's name to its numerical value.

CM_ALLOCATE_FAILURE_NO_RETRY	1	(X'01')
CM_ALLOCATE_FAILURE_RETRY	2	(X'02')
CM_CONVERSATION_TYPE_MISMATCH	3	(X'03')
CM_DEALLOCATED_ABEND	17	(X'11')
CM_DEALLOCATED_ABEND_BO	130	(X'82')
CM_DEALLOCATED_ABEND_SVC	30	(X'1E')
CM_DEALLOCATED_ABEND_SVC_BO	131	(X'83')
CM_DEALLOCATED_ABEND_TIMER	31	(X'1F')
CM_DEALLOCATED_ABEND_TIMER_BO	132	(X'84')
CM_DEALLOCATED_NORMAL	18	(X'12')
CM_DEALLOCATED_NORMAL_BO	135	(X'87')
CM_OK	0	(X'00')
CM_OPERATION_INCOMPLETE	35	(X'23')
CM_OPERATION_NOT_ACCEPTED	37	(X'25')
CM_PARAMETER_ERROR	19	(X'13')
CM_PIP_NOT_SPECIFIED_CORRECTLY	5	(X'05')
CM_PRODUCT_SPECIFIC_ERROR	20	(X'14')
CM_PROGRAM_ERROR_NO_TRUNC	21	(X'15')
CM_PROGRAM_ERROR_PURGING	22	(X'16')
CM_PROGRAM_ERROR_TRUNC	23	(X'17')
CM_PROGRAM_PARAMETER_CHECK	24	(X'18')
CM_PROGRAM_STATE_CHECK	25	(X'19')
CM_RESOURCE_FAILURE_NO_RETRY	26	(X'1A')
CM_RESOURCE_FAILURE_RETRY	27	(X'1B')
CM_RESOURCE_FAILURE_RETRY_BO	134	(X'86')
CM_RESOURCE_FAIL_NO_RETRY_BO	133	(X'85')
CM_SECURITY_NOT_VALID	6	(X'06')
CM_SVC_ERROR_NO_TRUNC	32	(X'20')
CM_SVC_ERROR_PURGING	33	(X'21')
CM_SVC_ERROR_TRUNC	34	(X'22')
CM_SYNC_LVL_NOT_SUPPORTED_LU	7	(X'07')
CM_SYNC_LVL_NOT_SUPPORTED_PGM	8	(X'08')
CM_SYSTEM_EVENT	36	(X'24')
CM_TAKE_BACKOUT	100	(X'64')
CM_TPN_NOT_RECOGNIZED	9	(X'09')
CM_TP_NOT_AVAILABLE_NO_RETRY	10	(X'0A')
CM_TP_NOT_AVAILABLE_RETRY	11	(X'0B')
CM_UNSUCCESSFUL	28	(X'1C')

Chapter 2. APPC/APPN Sense Data: Introduction

If you run low-level APPC traces (see the problem determination guide for your specific APPC product for information on how to run traces), you can get access to detailed sense data that can help you determine the cause of a particular problem.

Sense data is an SNA-defined encoding of error information. In LU 6.2 products, sense data can be found as part of internal SNA flows, error logs, error messages, and traces. In IBM's OS/2 Communications Manager, for example, it is even returned in the verb control block of an Allocate call (when using the native APPC API) when it fails because of an ALLOCATION_FAILURE condition. The sense data is a 4-byte field (as shown in the following figure) that includes:

- a 1-byte category value
- a 1-byte modifier value
- 2 bytes of sense-code specific information.

```
<--- Byte 0 ---><--- Byte 1 ---><--- Byte 2 ---><--- Byte 3 --->

<---Category---><---Modifier---><----Sense-code specific------->
                                  information
<----------Sense Code---------->

<------------------Sense Data----------------------------------->
```

Together, the category byte 0, the modifier byte 1, and the sense-code specific bytes 2 and 3 hold the sense data defined for the exception condition that has occurred. The sense codes are discussed on the following pages. For information about sense data not documented here, refer to *SNA Formats*, IBM document number GA27-3136, or *SNA Network Product Formats*, IBM document number LY43-0081.

Note: This information can also be found on the diskette accompanying the book *CPI-C Programming in C: An Application Developer's Guide to APPC*, published by McGraw-Hill, New York, NY, ISBN 0-07-911733-3.

X'08' Request Reject

This category indicates that the request was delivered to the intended component, and that it was understood and supported. However, the request was not executed.

0801xxxx Resource Not Available

Explanation: The LU, PU, link station, or link specified in an RU is not available. For example, this sense data can occur when a workstation is executing both of the following:

- A 3270 network connection directly to the host
- A link to an SNA gateway which is configured to support that workstation.

If the gateway sends an ACTPU to the workstation before the host, the workstation will reject the ACTPU from the host with this sense data. The 3270 emulator component logs this error as type 0042, subtype 000002B2 in IBM's OS/2 Communications Manager error log. Unless this situation is corrected, the host will continue to send the ACTPU and each negative response will cause an error to be logged.

Operator Response: To correct this problem, do one of the following:

- Change the configuration at the SNA gateway so that it no longer supports the workstation.
- Change the configuration at the workstation so that its 3270 network connection is to the gateway.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

- On CINIT: a destination LU rejected the CINIT because it could not support the logical unit type (such as an LU type 2) of the originating LU.
- On INIT_SELF: a destination LU rejected an INIT_SELF of the SNASVCMG session. The SNASVCMG mode session was being activated with an INIT_SELF because the partner LU definition in the OS/2 Communication Manager node definition file for a dependent LU partner was set with parallel_session_support(YES). Changing to parallel_session_support(NO) fixes this.

0009

The LU is not available because it is not ready to accept sessions. That LU is still in pending-delete state because it has active sessions with one or more partners. However, no new sessions can be started.

0012

An APPN connection cannot be established because the local node has no available integers to represent a new transmission group (TG). This error is associated with XID processing.

0805xxxx Session Limit Exceeded

Explanation: The requested session cannot be activated because one of the network addressable units (NAUs) has reached its session limit (such as the LU-LU session limit or the LU-mode session limit). This sense code applies to SNA ACTCDRM, INIT, BIND, and CINIT requests.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

For example, when VTAM is re-IPLed, all of its session limits are set to zero, pending an Initialize Change-Number-of-Sessions (CNOS) operation. Connected nodes may see this sense data if they were previously connected to VTAM and have not issued the CNOS operation again after it re-IPLed.

0001

If accepted, the BIND request would prevent either the receiving LU or the sending LU from activating the number of contention winner sessions to the partner LU that were agreed on during a CNOS procedure.

It is possible that two different LUs with the same name in the network attempted to set up sessions with the same third, target LU. The target LU rejects the BIND request issued by the later, duplicate named LU with this sense data. Each LU across every computer in an SNA network must have a unique, fully-qualified LU name.

Operator Response: Ensure that each LU has a unique, fully-qualified LU name in the same network.

0002

If the BIND request had been accepted, it would have caused the XRF-backup session limit to be exceeded.

0003

If the BIND request had been accepted, it would have caused the XRF-active session limit to be exceeded. The session limit for XRF-active sessions is 1. An XRF-active BIND is valid only if there are no XRF-active or XRF-backup sessions with the receiving secondary LU (SLU).

0004

For an independent LU, if the BIND request had been accepted, it would have caused the system-defined maximum number of sessions allowed for any LU to be exceeded for this LU.

0005

The intermediate session router is unable to create a session connector control block. The pool of session connectors is saturated with active sessions and with pending active sessions for which the queue bit was set in the BIND request; the BIND should not be retried.

0006

The intermediate session router is unable to create a session connector control block. The pool of session connectors is saturated with active sessions and with pending active sessions for which the queue bit was not set in the BIND request; the BIND should be retried.

0008

For a dependent LU, if accepted, the BIND request would cause the session limit to be exceeded.

0009

If the request had been accepted, it would have caused the primary logical unit (PLU) session limit to be exceeded.

000A

If the request had been accepted, it would have caused the secondary logical unit (SLU) session limit to be exceeded.

000B

The request was rejected because a session already exists between the same LU pair, and at least one of the LUs does not support parallel sessions.

000C

An LU-LU session was not set up because a session already exists between the SLU and the session-controller PLU.

0806xxxx Resource Unknown

Explanation: The request contained a name or address not identifying a PU, LU, SSCP, link, or link station known to the receiver or the sender.

In an interconnected network environment, this sense code may be set by an SSCP in whose subnetwork and domain the LU was expected to reside. It is not set by an SSCP that is only an intermediary on the session-setup path. A gateway SSCP examines the Resource Identifier control vector in a session setup request (for example, CD-Initiate), to determine whether the LU is in the SSCP's subnetwork and domain.

If bytes 2 and 3 appear to be out of range, they correspond to a CICS DFH message number. See entry 'yyyy' below. Otherwise, bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

The resources identified in an SNA Address List (X'04') management services (MS) common subvector are unknown to the PU receiving the request.

When this sense data flows in a -RSP(NMVT), the referenced X'04' subvector is the one that was present in the corresponding request NMVT. When this sense data flows in a Sense Data (X'7D') management services (MS) common subvector, the referenced X'04' subvector is present with the X'7D' subvector in the same major vector.

0002

Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0004

The indicated resources in the accompanying Name List (X'06') subvector are unknown to the control point to which the request containing the subvector was routed. Names in the hierarchy below the level of the first unknown resource are not examined by the control point.

0006

For a dynamic reconfiguration DELETE, MOVE, or REPLACE operation, the resource to be dynamically reconfigured could not be found.

0007

The LU address in bytes 8-9 of the Request Network Address Assignment (RNAA) request unit type X'4' is already in the free pool.

0008

For a dynamic reconfiguration DELETE, MOVE, or REPLACE operation, the network addressable unit (NAU) name in the Request Network Address Assignment (RNAA) request unit does not correspond to the resource identified by the element address in the RNAA.

0009

The SSCP(OLU) cannot identify the SSCP(DLU), and default SSCP rerouting is not enabled.

000A

The configuration identifier specified in a management services command is not recognized by the DLC manager at the receiving node.

0011

An unknown OLU name was specified in the request. For example, this situation can occur when a local node is using an LU that is unknown to the subarea. Ensure that all local LUs that set up sessions with a subarea host are configured at the host.

With the OS/2 Communications Manager, this can occur when using CPI-C, but failing to set the APPCLLU environment variable to the correct local LU name. The host will do CNOS to negotiate the session limits for the LU it is supposed to be using. The APPC software in the local OS/2 computer then sends a BIND request with the wrong workstation LU name, and the host replies with this sense data.

0012

An unknown destination logical unit (DLU) name was specified in the request.

0013
An unknown secondary logical unit (SLU) name was specified in the request.

0014
An unknown primary logical unit (PLU) name was specified in the request.

0015
An unknown origin logical unit (OLU) address was specified in the request.

0016
An unknown destination logical unit (DLU) address was specified in the request.

0017
An unknown secondary logical unit (SLU) address was specified in the request.

0018
An unknown primary logical unit (PLU) address was specified in the request.

0021
The session-initiation request specified that the receiving SSCP is the SSCP having the DLU in its domain, but the DLU is unknown to the receiving SSCP.

For VTAM, this sense data indicates that a link with a downstream node is defined incorrectly. For example, this can occur when VTAM thinks it link to an adjacent node is an APPN-level link, but the adjacent node is treating the link as a LEN-level link. If the adjacent node had treated the link as APPN-level, it would have sent an APPN Locate to VTAM prior to sending its BIND request. The result would have been that the RSCV used to route the BIND request would include an additional “Interchange TG” hop, which indicates to VTAM that the destination LU is actually located in the subarea network.

If VTAM receives a BIND request over an APPN-level link and the RSCV ends on that VTAM (indicating that the DLU should be on that VTAM), then VTAM will reject the BIND request if the DLU is not owned locally. However, prior to rejecting the BIND request, VTAM will send a DSRLST into the subarea. If the DLU is found in the subarea, this this sense data is set to inform the OLU that the DLU does not reside on this node.

0022
The originator of the request is unknown to the receiver.

0023
The destination of the request or response is unknown to the sender.

0024

An unknown LU1 name was specified in the request.

0025

An unknown LU2 name was specified in the request.

0026

The SSCP does not have a session with the boundary function PU of an independent LU.

0027

The PU associated with a switched secondary logical unit (SLU) is unknown. Session setup processing for the switched SLU cannot proceed.

0028

NAU1 network address is unknown.

0029

NAU2 network address is unknown.

002A

The NAU name in the CONTACT or ACTLU does not correspond to the resource at the target address.

002B

The TG (link) being activated is unknown. The local or network APPN topology information may be incorrect. This can occur because of one of the following conditions:

- The local node is trying to use an “activatable on demand” link, but could not activate the link. This could happen for any of the reasons a link activation can fail: the link was defined wrong, the remote node is not powered on, and so on.
- An “activatable on demand” link is defined in the reverse direction from where the local node is trying to go. To correct this, either make your local link “activatable on demand,” or change the configuration at the remote node so that it is not “activatable on demand.”
- There is an inconsistency among the APPN topology databases of the network nodes in the APPN network. One way to spot inconsistencies, using the OS/2 Communications Manager for example, is to issue the DISPLAY_APPN verb, requesting Topology Information. See if there are any links shown as active that are not actually active. To correct this condition in the local workstation, stop and re-start the OS/2 Communications Manager. If this condition persists, there may be an internal APPN software error somewhere in the network.

002C

The identification in either the Node Identification field or the control point (CP) name in a received XID3 is different from what the local node was configured to expect. The CP name configured in the link definition does not match what was expected. This error is associated with XID processing.

002F

The destination resource was not found on this node during a session activation attempt.

0030

The adjacent node was not identified during CP-CP session activation.

yyyy

If bytes 2 and 3 appear to be out of range, these two bytes correspond to a CICS DFH message number. An Allocate call failed when trying to BIND request to a CICS application through VTAM. VTAM has not yet acquired the local LU. The BIND request should succeed after forcing an acquire. Assure that your local LU has a corresponding TCT entry.

0809xxxx Mode Inconsistency

Explanation: The requested function cannot be performed in the present state of the receiver.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

Note: VTAM APPC will erroneously reject an inbound BIND request with this sense data if this is the first session for this LU (no CNOS yet) and the mode name is not SNASVCMG. Some APPC resources will not react properly (for example, they will hang) when receiving this sense data. The expected sense data is `X'08050000'`.

For VTAM, apply JR06063 plus the appropriate VTAM fix:

Release 301:	UY82095
Release 303:	UY82094
Release 401:	UY82096
Release 431:	UY82097

0001 - 000D

Set aside for implementation-specific use, and will not be otherwise defined in SNA. See the platform documentation for details of usage.

000A

The partner data link control (DLC) is not activated. In the OS/2 Communications Manager a long timeout (for example, 30 seconds) occurs before the verb returns. Possible reasons this occurs are:

- The target computer is not powered on.
- The target data link control (DLC) is not activated.
- The target MAC address is wrong.
- The maximum number of links is exceeded.
- All links configured in the local workstation are incoming.
- Different service access points (SAPs) are defined.

0039

CP Transaction Error. A CP Capabilities (X'12C1') generalized data stream (GDS) variable request was sent indicating conversation complete or without change direction (for example, CEB or ¬CD) or CP Capabilities reply was sent indicating conversation not yet complete (for example, ¬CEB).

Generically, this sense data value is issued whenever the LU 6.2 transaction to request and provide a CP Capabilities GDS variable does not take place as designed. This is most easily detected by looking at the CEB and CD bits. The *SNA APPN Architecture Reference* document is a bit more general; it tells implementers that this sense data is also to be used when any other transaction-related errors are discovered. In reality, the REQ_CP_CAP_TP internal transaction program issues some RECEIVE_AND_WAIT verbs, and once everything is received, it checks for a primary return code of DEALLOCATE_NORMAL and a verb count of 2. If either one is not true, it issues the 08090039 sense data value.

003A

A null XID was received when a link activation XID3 was expected. Specifically, a null XID was received when an XID3 with its Exchange State indicators set to "prenegotiation," "negotiation proceeding," or "Exchange State indicators not supported" was expected. This error is associated with XID processing.

003B

A null XID was received when a nonactivation XID3 was expected. This error is associated with XID processing.

003C

An XID3 with the Exchange State indicators set to "prenegotiation" was received when either of the values "negotiation proceeding" or "Exchange State indicators not supported" was expected. This error is associated with XID processing.

003D

A nonactivation XID3 was received when a null XID or link activation XID3 was expected. This error is associated with XID processing.

003E

A link activation XID3 was received when a null XID or nonactivation XID3 was expected. This error is associated with XID processing.

003F

A secondary link station attempted to initiate a nonactivation XID exchange when this option is not supported on the TG. This error is associated with XID processing.

0040

A mode-setting command was received and was either not expected or invalid for the receiving node. For example, SNRME was received when SNRM was expected. This error is associated with XID processing.

0041

Negotiation-proceeding XID3 was received when prenegotiation XID3 was expected. This error is associated with XID processing.

0042

On an asynchronous balanced mode (ABM) TG on which secondary-initiated nonactivation exchanges are supported, the adjacent node initiated a nonactivation exchange without explicitly indicating that it was doing so in the ABM Nonactivation XID Initiator indicator in XID3. This error is associated with XID processing.

0045

An XID3, indicating that the sender does not support the Exchange State indicators, was received when the sender had previously indicated support for this field in XID3. This error is associated with XID processing.

0046

An XID3, indicating that the sender supports the Exchange State indicators, was received when the sender had previously indicated that it does not support this field in XID3. This error is associated with XID processing.

0047

An XID was received after receipt of a mode-setting command, but before the completion of the mode-setting sequence, that is, before RR, RNR, or an I-frame with the Poll bit set has been sent by the node with the primary link station after it has received UA in response to its mode-setting command. Once a node sends one of these mode-setting commands, no additional XIDs may be sent. This error is associated with XID processing.

0048

An NRM primary link station received an unsolicited XID from an NRM secondary link station. This error is associated with XID processing.

0049

An XID3 containing the XID Negotiation Error (X'22') was received by this node. This error is associated with XID processing.

For example, when using the OS/2 Communications Manager in the local node, this can occur when there are no link stations available on the local DLC adapter to connect a link to the adjacent computer. In one scenario, a user had the “max link stations” on their DLC profile set to 2, with the percentage reserved for incoming sessions equal to 50%. Only one link station was reserved for connecting out, and once that was already in use, subsequent requests by other partners were rejected with this sense data. Increasing the number of link stations higher than 3 corrected the situation.

004E

A node with an NRM secondary link station attempted to initiate a nonactivation XID exchange with an XID. This error is associated with XID processing.

0054

The adjacent node is not the node type (that is, network node or end node) that the receiving node was configured to expect. The received negotiation-proceeding XID3 indicated that the adjacent node is an APPN end node while this node expected the adjacent node to be an APPN network node, or vice versa. This error is associated with XID processing.

0055

The virtual routing node named in the received XID3 in the TG Identifier (X'80') subfield of the TG Descriptor (X'46') control vector is not defined on the receiving port. This error is associated with XID processing.

080Axxxx Permission Rejected

Explanation: The receiver has denied an implicit or explicit request of the sender.

When sent in response to a BIND request, it implies either that the secondary LU will not notify the SSCP when a BIND request can be accepted, or that the SSCP does not recognize the NOTIFY vector X'0C'. (See the X'0845' sense code for a contrasting response.)

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

An SSCP has denied permission to establish a session through its gateway resources; the receiving SSCP should not attempt to reroute the request to another SSCP.

0002

An SSCP has denied permission to establish a session through its gateway resources; the receiving SSCP should not attempt to reroute the request to another SSCP.

080Cxxxx Procedure Not Supported

Explanation: A internal SNA procedure (Test, Trace, IPL, REQMS type, MS major vector key) specified in a request/response unit (RU) is not supported by the receiver.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0005

The management services (MS) major vector key is not supported by the receiver. This may occur for one of these reasons:

- The local node sent an NMVT to the host using its active SSCP-PU session. At the host, one of two things occurred:
  - No network management application available (such as NetView) to handle the NMVT.
  - The network management application on the host does not support the management services (MS) major vector key sent by APPC.

    When the host replied that it could not handle the NMVT, APPC on the local computer recorded this sense data.
- NetView or a NetView application on the host sent an NMVT request to local node, but it was not recognized by the local node.

  For more information about NS headers for NMVT, refer to *SNA Formats*, IBM document number GA27-3136.

0006

The management services (MS) major vector is identified as one that contains a command, but the receiver does not recognize or support the command subvector. A control subvector is missing or not supported. Products that are not MS focal point products, do not know how to process a X'63', so they send an SNACR rather than a X'64' reject.

If the command subvector is identified, but an additional required subvector is missing, refer to X'086C' sense code in *SNA Formats*, IBM document number GA27-3136.

0012

Application generalized data stream (GDS) variable in a Multiple-Domain Support Message Unit (MDS_MU) not supported.

0013

MDS message type not supported. Receiving node does not support the MDS message type in this MDS_MU.

01xx

A subvector in a management services (MS) major vector is identified as one inside which the receiver requires one of several supported subfields, but none of these subfields is present. Byte 3 contains the key (xx) of the subvector.

080Exxxx NAU Not Authorized

Explanation: The requesting network addressable unit (NAU) does not have access to the requested resource.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0002

A gateway Type 4 (T4) node received a dynamic dump request from an SSCP that is not in the native network of the gateway T4 node.

0006

An APPN network node received a Register from an unauthorized APPN end node.

0009

An APPN network node received an APPN Delete request from an unauthorized APPN end node.

000B

A Locate/CDINIT was received from a node that is not defined as a client APPN end node. This can be detected by either APPN directory services or session services. For example, a BIND request came into an APPN network node that is not the APPN network node server of the BIND-sending EN. The APPN network node is unable to handle this type of request.

Operator Response: Define that APPN network node as your NN server, or send the BIND request to the currently defined NN server.

080Fxxxx Security Mismatch

There is mismatch with the supplied used ID or password and what was expected by the partner. This can indicate an attempt at a security violation.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

End User Not Authorized Explanation: This indicates that the BIND request is correct, but the primary LU has not been authorized to activate the requested session.

6051

Access Security Information Invalid. The request specifies an Access Security Information field that is unacceptable to the receiver; for security reasons, no further detail on the error is provided. This sense data is sent in FMH-7 or UNBIND.

The partner LU rejected the user ID or password received on an incoming Attach, or there is a mismatch in the LU-LU security.

For example:

- The remote location is using OS/2 Communications Manager and the TP definition specifies conversation_security(YES). The remote Attach Manager searched its list of valid user IDs and passwords, but did not find the received combination.
- When an application sends an Attach to the AS/400 with security(NONE) (that is, with no user ID and password), the AS/400 looks for a default user profile to assign the request to. If the default user profile is not set up, the AS/400 rejects the request.
- For mismatches in LU-LU security, trace the line flows and look in the BIND for the presence of structured data subfield `X'11'`.
  1. If subfield `X'11'` is present, then the workstation is attempting to do LU-LU verification and the host is rejecting it. This is probably due to either 1) being configured to not accept LU-LU verification from the workstation, or 2) not being able to access the LU-LU password.

     Check the VTAM definitions to see if LU-LU verification is allowed. To do this, check if the VTAM APPL statement has SECACPT=NONE, which says “don't allow security fields in incoming Attaches.” If the application (LU) really wants RACF to check a user ID from a partner, this value should be SECACPT=(ALREADYV or CONV or AVFV or PERSISTV).

     If LU-LU verification is allowed, check the ACF2 definitions for a resource named vtamnetid.vtamlu.cm2lu in class APPCLU. Also, make sure that the class is active. Or, turn off LU-LU verification at the workstation.
  2. If subfield `X'11'` is not present, then the workstation is not attempting LU-LU verification and the host is requiring it. In this case, VTAM is probably configured to require LU-LU verification and workstation is not configured for LU-LU verification.

     Once, again check the VTAM definition.

Programmer Response: The Allocate call issued by the local program has a wrong user ID or password parameter, or the remote configuration needs to be changed. Look at the verb control block being used for this verb. Verify that it specifies valid user ID and password parameters.

Operator Response: If the problem is with the remote configuration, assure that the user ID and password combination has been correctly defined there. User IDs and passwords are two of the few APPC configuration fields that are case-sensitive. Be sure that the combination of uppercase and lowercase letters matches those specified in the program.

Also assure that if these are required (see the partner's TP definition), they will be accepted by the partner LU (see the remote location's definition of its partner LU).

If the partner is using the OS/2 Communications Manager, look at its error log. Find an error log of type 001A; its subtype indicates why this Attach was rejected.

0812xxxx Insufficient Resource

Explanation: Receiver cannot act on the request because of a temporary lack of resources.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0006

A local LEN node is acting as a independent LU, but the target LU supports only dependent logical units (LUs).

0007

Insufficient resources are available for LU address allocation.

000A

An APPN network node does not have adequate resources to perform the Register. For example, the available APPN directory capacity is exceeded.

000D

Insufficient buffers exist to activate a session.

0010

The control point (CP) does not have adequate resources to process a Locate generalized data stream (GDS) variable request. The contention loser's CP-CP session is deactivated.

0019

Insufficient storage to conduct an XID exchange.

001A

Insufficient storage to activate a TG or a link.

001B

Insufficient storage to activate a token-ring connection.

08130000 Bracket Bid Reject—No RTR Forthcoming

Explanation: BID (or BB) was received while the first speaker was in the in-bracket state, or while the first speaker was in the between-brackets state and the first speaker denied permission. RTR will not be sent. For LU 6.2, this is the only setting defined.

08140000 Bracket Bid Reject—RTR Forthcoming

Explanation: BID (or BB) was received while the first speaker was in the in-bracket state, or while the first speaker was in the between-brackets state and the first speaker denied permission. RTR will be sent. For LU 6.2, this is the only setting defined.

0815xxxx Function Active

Explanation: A request to activate a network element or procedure was received, but the element or procedure was already active.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0004

A BIND request was received from a Type 2.1 node when the session is already active. For example, the LFSID is in use. The receiver rejects the BIND request.

0007

A session activation request was received by an APPN network node or an APPN end node to activate a CP-CP session that was already active.

0817xxxx TG Not Activated

Explanation: The TG (link) could not be activated.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001

The TG requires operator intervention to activate it. For example, the link station is not defined to support automatic activation. This error is associated with ACTIVATE_ROUTE processing.

0002

The TG number has been changed by a nonactivation XID3 from the TG number negotiated during link activation. This error is associated with ACTIVATE_ROUTE processing.

08190000 RTR Not Required

Explanation: The receiver of Ready-To-Receive has nothing to send.

081Cxxxx Request Not Executable

Explanation: The requested function cannot be executed, because of a permanent error condition in the receiver.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0103

The remote node is not responding to polling requests. The remote node may be powered off or the hardware may be functioning incorrectly.

081E0000 Session Reference Error

Explanation: The request contained reference to a half-session that either could not be found or was not in the expected state (generally applies to network services requests).

0821xxxx Invalid Session Parameters

Explanation: Session parameters were not valid or not supported by the half-session whose activation was requested.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0002

indicates that the specified mode name was not recognized by the control point (CP). If VTAM generates this sense data, it means the VTAM logmode table for the LU in question does not have the mode name defined. If the LU has been dynamically created by VTAM, then the default logmode table (ISTINCLM) does not contain the mode entry.

0824xxxx Logical Unit of Work Terminated

Explanation: The current unit of work has been terminated; when sync point protocols are in use, both sync point managers are to revert to the previously committed sync point. For LU 6.2, this sense data is sent only on FMH-7.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

indicates for LU 6.2, Backout Initiated—No Resync in Progress. A transaction program or its LU has initiated backout. The protected resources for the distributed logical unit of work are to be restored to the previously committed sync point. When sent in reply to a PS header, no resync in progress means that all resources in the transaction subordinate to the backout sender have backed out.

08260000 FM Function Not Supported

Explanation: Function management request not supported. A function requested in an FMD RU is not supported by the receiver.

For example, given a connection from APPC/MVS to OS/2, VTAM or APPC/MVS is requesting the OS/2 Communications Manager to do something it doesn't support. All the reasons for this should be reported via sense code 1003nnnn instead. Get a trace of the OS/2 Communications Manager side to see the details of that sense data.

0835xxxx Invalid Parameter (with Pointer Only)

Explanation: The request contained a fixed- or variable-length field whose contents are invalid or not supported by the network addressable unit (NAU) that received the request.

xxxx

Bytes 2 and 3 contain a two-byte binary count that indexes (0-origin) the first byte of the fixed- or variable-length field having invalid contents.

The error usually indicates that the BIND request was rejected by APPC because of a syntax, state, or semantic error and points to an offset in the RU field of the invalid parameter. If possible, look at the BIND request in a trace or message; the partner LU is wrong, or the spelling of the partner LU or mode name is wrong.

For example, if a BIND request is rejected with a sense data of X'08350045', it implies that the field at offset X'45' is in error. If the field at this offset is an LU name, is could indicate that duplicate LU names are present in the network. If the field is a mode name, it could indicate a misspelled mode name was supplied on an Allocate call, or mode name definition was missing or misspelled at either the local or remote computer.

With the OS/2 Communications Manager, you can make a simple fix by specifying inbound_plu(YES), allowing any incoming partner LU name.

0018

The partner LU rejected the value for “sync level” or “parallel session support.” For example, if you are connecting to IMS, it is likely that the problem is the parallel session support bit. To solve this, define a partner LU for use with the IMS adapter, and specify no parallel session support.

This sense data is not used to report an invalid value in a management services (MS) major vector. If the invalid value occurs in a formatted management services (MS) subvector, sense code X'086B' is used. If it occurs in an unformatted subvector, sense code X'0870' is used.

0839xxxx LU-LU Session Being Taken Down or LU Being Deactivated

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

During session-initiation processing, a session-termination request has caused the LU-LU session to be taken down.

0002

A Request Network Address Assignment (RNAA) request unit (type 3) was received for a session during the process of session deactivation. The RNAA should be retried.

0003

SSCP detected that this session should no longer exist and requested its termination. For example, a BFSESSINFO was received reporting a subject LU address that the SSCP believed already belonged to an other-domain resource.

0004

CP Architected-TP Send Failure. This can occur, for example, when an APPN network node attempts to issue a broadcast search to a particular node and the node does not respond within a timeout period. An UNBIND request is sent to the problem node and the session is deactivated.

Deactivate both conwinner and conloser CP-CP sessions.

083Axxxx LU Not Enabled

Explanation: At the time an LU-LU session initiation request is received at the SSCP, at least one of the two LUs (although having an active session with its SSCP) is not ready to accept CINIT or BIND requests.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

The primary logical unit (PLU) is not enabled.

0002

The secondary logical unit (SLU) is not enabled.

This sense data is set when VTAM has activated the LU (that is, the SSCP-LU session exists), but the device is not enabled. The means the PU that supports the dependent LU is active, but the terminal itself has not powered on yet. (For emulated dependent LUs, this probably means the LU itself has not initialized itself.) You should recycle the dependent LU.

083Bxxxx Invalid PCID

Explanation: The received procedure correlation identification (PCID) for a new session duplicated the PCID assigned to another session, or the received PCID intended as an identifier for an existing session could not be associated with such an existing session, or an error was detected in the format of the received PCID.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

The PCID contained in CDINIT (Initiate or Queue), INIT-OTHER-CD, or CDTAKED duplicates a PCID received previously in one of these requests.

0002

The received fully-qualified PCID duplicated one assigned to another session.

083Exxxx Implementation-Defined Retry Limit Exhausted

Explanation: The implementation-defined limit on XID exchanges was exceeded before link activation completed.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001

Limit for link activation XID exchanges exceeded.

For example, the maximum BTU (MAX_BTU) size of the host is less that the MAX_BTU size of a connected workstation. The MAX_BTU size is the largest segment of data that can be sent or received over a link. The host will keep trying to go into Negotiation_Proceeding when the two sides are in disagreement over the MAX_BTU size.

0002

Limit for nonactivation XID exchanges exceeded.

0840xxxx Procedure Invalid for Resource

Explanation: The received RU is not supported in the receiver for this type of resource. For example:

- SETCV specifies boundary function support for a type 1 node but the capability is not supported by the receiving node.
- The PU receiving an EXECTEST or TESTMODE is not the primary PU for the target link.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0007

Resource Not Found. An APPN directory request (Delete or Find) could not be satisfied because the specified entry does not exist in the receiver's directory and the resource cannot be found in the network.

The target LU may be owned by an APPN network node server that cannot be contacted, or it might not exist in the network at all. For example,

- The network node server has lost its links to the other network nodes in its APPN network.

- A second NN has connected to an EN that already has an NN server. If the second connection to the EN had been defined as a LEN connection, or the EN's connection to the second NN had been defined so that CP-CP session support had not been requested, the CP LU would have been put in the APPN directory when the link came up. However, since it is a multi-tail EN, the second NN did not put it in its directory; it will not answer network broadcast searches for the EN positively. You want only the NN server to answer so that all the EN's links are returned and considered for route calculation. If your node had any connectivity to the NN server, the Allocate call would have worked from your NN, probably going across the direct link.

  Since ENs do not have to worry about answering network broadcasts, it puts any contacted CP LU in its directory database, so it could find the NN when it did an Allocate call. Of course, the NN then used the existing session to do its Allocate call.

0008

Directory Entry Cannot Be Deleted. An APPN network node received an APPN Delete request with a delete entry condition indicating that the entry can be deleted only if it has no subordinate entries. Since the entry does have subordinate entries, the Delete is rejected.

000C

Conflicting Entry Type on Delete. The APPN Delete request attempted to delete a home entry. For example, one defined at the receiver by its own network operator facility (NOF).

08460000 ERP Message Forthcoming

Explanation: The received request was rejected for a reason to be specified in a forthcoming request.

084Bxxxx Requested Resources Not Available

Explanation: Resources named in the request, and required to honor it, are not currently available. It is not known when the resources will be made available.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

6031

Transaction Program Not Available—Retry Allowed. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a transaction program that the receiver is unable to start. Either the program is not authorized to run or the resources to run it are not available now. The condition is temporary. The sender is responsible for subsequent retry. This sense data is sent only in FMH-7.

The partner LU rejected the incoming Attach because it could not start the specified program immediately.

If the partner LU is associated with the APPC Attach Manager in the OS/2 Communications Manager, some possible causes of this sense data are:

- The Attach Manager associated with the partner LU is stopped.
- The incoming_allocate_timeout defined in the TP definition at the partner LU is zero or is too short. This only applies if the partner TP is configured as Queued.
- The incoming_allocate_queue_depth configured in the TP definition at the partner LU is too small. This only applies if the partner TP is configured as Queued.
- At the remote location, the incoming Attach was queued, waiting to be processed. The intended program was remotely started, but ended before it could process the queued incoming Attach.

Programmer Response: Retry the allocation request. However, to avoid congesting the network with attempted allocation requests, your local program should pause or wait for a keystroke before retrying the conversation.

Operator Response: Start the APPC Attach Manager, increase the incoming_allocate_timeout, increase the incoming_allocate_queue_depth in the configuration at the remote location, if appropriate.

At the remote location, the remote program may have ended before issuing an Accept_Conversation (CMACCP) call (or equivalent) to receive the queued Attach. If this action is unexpected, check the following conditions in the remote location:

- The remote program might be failing to issue the Accept_Conversation (CMACCP) calls correctly.
- The remote program might be issuing Accept_Conversation (CMACCP) calls with an incorrect tp_name.
- The filespec field configured at the remote location could be incorrect.

If the remote location is using the OS/2 Communications Manager, look at its error log. Find an error log of type 0020; its subtype indicates why this Attach was rejected.

084C0000 Transaction Program Not Available—No Retry

Explanation: The partner LU rejected the incoming Attach because it could not start the specified program. Receiver cannot act on the request because resources required to honor the request are permanently unavailable. The sender should not retry immediately because the situation is not transient.

For LU 6.2, an Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a transaction program that the receiver is unable to start. The condition is not temporary. The sender should not retry immediately. This sense data is sent only in FMH-7.

Examples:

- The intended program does not exist at the remote location.
- If the remote location is using the OS/2 Communications Manager, its TP definition may be configured with the wrong filespec parameter, or the filespec for the TP may be spelled wrong.
- The operating system or communications subsystem at the remote location has exhausted its available resources and is unable to start the remotely attachable program.

  If the remote location is using the OS/2 Communications Manager, the APPC Attach Manager at the remote location may have failed to start the program because of an OS/2 error on its internal DosExecPgm or DosStartSession function call. Any of the non-zero OS/2 return codes from these function calls may cause this situation. For example, the remote location is running OS/2 Communications Manager and the remotely attachable program is attach-manager-started. If there is insufficient memory for the Attach Manager to start the program, the incoming Attach will be rejected with this sense data. This could similarly occur if the remote location has used all of its processes or screen groups.

Programmer Response: Do not retry the allocation request.

Operator Response: Ensure that the intended program is defined correctly at the remote location, and that it is compiled and linked correctly for its operating system. If the remote location is using the OS/2 Communications Manager, look at its error log for more information.

08500001 Link Activation Limit Reached

Explanation: The specified TG was not activated because the maximum number of active link stations allowed on this port has already been reached. For example, this can be caused when no ports are available for a connection network.

08520001 Duplicate Session Activation Request

Explanation: A second BIND request has been received from a peripheral node PLU while the session was still in the activation process.

0857xxxx SSCP-LU Session Not Active

Explanation: The SSCP-LU session, required for the processing of a request, is not active; for example, in processing REQECHO, the SSCP did not have an active session with the target LU named in the REQECHO RU.

Bytes 2 and 3 following the sense code contain sense-code-specific information. Settings allowed are:

0000

No specific code applies.

0001
The SSCP-SLU session is in the process of being reactivated.

0002
The SSCP-PLU session is inactive.

0003
The SSCP-SLU session is inactive. If this sense data is received from VM/VTAM, it indicates that the LU is inactive. If it is received from APPC/VM LUs, it means that the gateway is inactive.

This sense data is returned when AVS is not running on VM. Restarting AVS will generally correct this problem.

0004
The SSCP-PLU session is in the process of being reactivated.

0005
The SSCP lost connectivity with the primary logical unit (PLU) after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0006
The SSCP lost connectivity with the secondary logical unit (SLU) after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0007
The selected ALS for the OLU is not in a state permitting LU-LU sessions to be set up using it. The condition is detected when the session request (BFINIT) was received, but, when the request was processed, the ALS was no longer in an active state. The session request is rejected.

0008
The selected ALS for the DLU is not in a state permitting LU-LU sessions to be set up using it. The condition is detected when the session request was being processed in the DLU domain and the ALS selected for the DLU is no longer in an active state. The session request is rejected.

0860xxxx Function Not Supported--Continue Session
Explanation: The function requested is not supported; the function may have been specified by a request code or some other field, control character, or graphic character in an RU.

Bytes 2 and 3 following the sense code contain sense-code-specific information. Settings allowed are:

nnnn

Bytes 2 and 3 contain a 2-byte binary count that indexes (0-origin) the first byte in which an error was detected. This sense data is used to request that the session continue, thereby ignoring the error.

0861xxxx Invalid COS Name

Explanation: The class-of-service (COS) name, either specified by the independent LU (ILU) or generated by the SSCP of the secondary LU (SLU) from the mode table is not in the “COS name to VR identifier list” table used by the SSCP of the primary LU (PLU).

Bytes 2 and 3 following the sense code contain sense-code-specific information. Settings allowed are:

0000

The class-of-service (COS) name was generated by the SSCP.

0001

The class-of-service (COS) name was generated by the independent LU (ILU).

0002

The class-of-service (COS) name generated by the Type 2.1 control point (CP) local to, or the Type 2.1 NNCP server for, the initiating logical unit is not in the class-of-service name definition table.

For example, this sense data occurs when VTAM sends a BIND request with a COS name that is not defined in an APPN network node on the route.

0003

The CD-Initiate (CDINIT) request or response contains a Session Initiation control vector that has class-of-service (COS) name fields that have not been properly specified.

0864xxxx Function Terminated

Explanation: The conversation was terminated abnormally. Other terminations may occur after repeated executions; the request sender is responsible to detect such a loop.

Bytes 2 and 3 following the sense code contain sense-code-specific information.

0000

Premature Conversation Termination. An active conversation was terminated abnormally. This sense data is sent only in FMH-7 or UNBIND. The partner program may have issued a Deallocate type(ABEND) call, or the partner program may have terminated (normally or abnormally) without explicitly terminating the conversation.

If the partner program is running on OS/2, this sense data would be seen locally if the partner program ended unexpectedly because of a protection fault.

On the AS/400, this sense data can be caused by the absence of a valid routing entry in a subsystem. If no specific subsystem device or remote location name has been entered, check to see if the subsystem QCMN is active. If QCMN is active, be sure the routing entry for APPC programs (PGMEVOKE) is present in the subsystem. If a subsystem other than QCMN is being used and a specific communications entry or remote location name entry is being used, be sure a routing entry with PGMEVOKE is specified in the same subsystem. In both cases, the fix is to add a routing entry in a subsystem.

When a routing entry is not available on the AS/400, this sense data is written to QSYSOPR. There may be a job log produced, but it will not have any program information or CL statements printed.

0001

System Logic Error—No Retry. A system logic error has been detected. No retry of the conversation should be attempted. This sense data is sent only in FMH-7 or UNBIND.

0002

Excessive Elapsed Time—No Retry. Excessive time has elapsed while waiting for a required action or event. For example, a transaction program has failed to issue a conversation-related protocol boundary verb. No retry of the conversation should be attempted. This sense data is sent in UNBIND when there is no chain to respond to; otherwise, it is sent in FMH-7.

086Cxxxx Required Control Vector or Subvector Missing

Explanation: One or more control vectors or management services (MS) subvectors that are required by the receiver to perform some function are missing from the received messages, or are not present in the required position.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

xx00

Byte 2 following the sense code contains the key (xx) of one of the control vectors or subvectors that is missing, or improperly positioned. Byte 3 is reserved (00).

See the X'080C0006' sense data for the condition in which the major vector key is recognized, but a subvector representing the function to be performed cannot be identified.

086Dxxxx Required Subfield Missing

Explanation: A control vector or management services (MS) subvector lacks one or more subfield keys that are required by the receiver to perform the function requested.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

xxyy

Byte 2 following the sense code contains the subvector key (xx) of the subvector lacking a required subfield, and byte 3 contains the subfield key (yy) of a missing subfield.

086Fxxxx Length Error

Explanation: A length field within an management services (MS) major vector structure is invalid, or two or more length fields are incompatible.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

Control vector length field is invalid or control vector fields are incompatible.

0002

The sum of the management services (MS) subvector lengths is incompatible with the MS major vector length.

xx05

Management services (MS) subvector length invalid. Byte 2 following the sense code contains the relevant subvector key (xx). (This is specified only if the sum of the subvector lengths is compatible with the major vector length.)

xx06

Subfield length invalid. Byte 2 following the sense code contains the subvector key (xx) of the management services (MS) subvector containing the invalid subfield length. (This is specified only if the sum of the subfield lengths is compatible with the subvector length.)

0007

Invalid or Incompatible length fields in an MDS_MU. The length field of an MDS_MU is incompatible with the sum of the lengths of the imbedded generalized data stream (GDS) variables, or one of the imbedded GDS variables has a length field with a value of less than 4.

0870xxyy Unformatted Subvector Value Invalid

Explanation: A value in an unformatted management services (MS) subvector, or in an unformatted portion of a partially formatted MS subvector, is invalid.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

xxyy

Byte 2 following the sense code contains the subvector key (xx) of the management services (MS) subvector containing the invalid value. Byte 3 contains a one-byte binary count that index the first byte which the invalid value falls.

0876xxxx Nonreversible Explicit Route Requested

Explanation: The explicit route number (ERN) used by the NC-ACTVR does not use the same sequence of transmission groups (in reverse order) as the ERN that should be used for the RSP(NC-ACTVR).

0001

Subarea Could Not Locate LU (?)

0877xxxx Resource Mismatch

Explanation: The receiver of a request detected a mismatch between two of the following:

- The definition of an affected resource
- The actual configuration
- The definition of the resource as implied in the request.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

002A

A session cannot be activated because the node does not support segment generation and the maximum link basic transmission unit (BTU) size is too small to satisfy a requirement on the minimum send RU size as defined for the session mode.

002B

A session cannot be activated because the node does not support segment reassembly and the maximum link basic transmission unit (BTU) size is too small to satisfy a requirement on the minimum receive RU size as defined for the session mode.

002C

A BFINIT session request was received from a primary logical unit (PLU) that is not in the same network as this SSCP, or a BFSESSINFO was received reporting a subject LU in another network. A likely cause is a NETID mismatch with your subarea.

087Dxxxx Session Services Path Error

Explanation: A session services request cannot be rerouted along a path of SSCP-SSCP sessions. This capability is required, for example, to set up a cross-network LU-LU session.

Bytes 2 and 3 contain sense-code-specific information that indicates the specific reason for not rerouting the request.

0000

No specific code applies.

0001

VTAM has searched all the SSCPs in the subarea network, but the fully-qualified LU name specified as the destination in the BIND request cannot be found. The sense data means “routing exhausted,” which implies that VTAM looked everywhere it could in the subarea and did not find the LU. For example, this can occur if the wrong NETID or partner LU name is defined in the local node. If the destination is more than one hop away, VTAM's adjsscp search will not find it.

There are many possible causes, but the most likely are:

- The LU name specified in the BIND request is incorrect (in other words, the LU just doesn't exist in the subarea).
- A VTAM adjacent SSCP table is wrong and the SSCP that actually owns the LU was not searched because it did not appear in the correct table.
- A cross-domain resource management (CDRM) path is down and that prevented the search from reaching the SSCP that actually owns the LU.
- This sense data is frequently seen when the host is unavailable for a 3270 emulation session.

An SSCP has attempted unsuccessfully to reroute a session services request to its destination via one or more adjacent SSCPs; this value is sent by a gateway SSCP when it has exhausted trial-and-error rerouting.

This code is used when SSCP rerouting fails completely. The remaining codes are used for failures to reroute to a particular SSCP. For example, they are associated with specific SSCPs when information about a rerouting failure is displayed in the node that was trying to reroute.

The sense data is generic, but VTAM typically generates a message with message number 894I or 895I that will contain additional sense data information about the real cause of the problem.

0888xxxx Name Conflict

Explanation: A name specified in an RU conflicts with a previous usage, or is unknown, or is known and does not have the required capabilities, or is a duplicate resource for the specified resource type. When a name conflict is detected, further name checking ceases; multiple name conflicts are not reported or detected.

Bytes 2 and 3 following the sense code contain sense-code-specific information. Settings allowed are:

0000

No specific code applies.

0001
The specified DLU real network name is known, but identifies a resource that is not LU-LU session capable.

0002
The specified DLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0003
The specified OLU real network name is known, but identifies a resource that is not LU-LU session capable.

0004
The specified OLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0005
Name translation was invalid; that is, a different LU name was returned with the same network ID as the original LU name.

0006
The specified DLU real network name is known, but is a duplicate resource.

0007
The specified DLU alias network name is known, but is a duplicate resource.

0008
The specified OLU real network name is known, but is a duplicate resource.

0009
The specified OLU alias network name is known, but is a duplicate resource.

000B
A cross-network DLU name is defined as a shadow resource, but shadow resources are not supported for cross-network sessions.

000C
Set aside for implementation-specific use, and will not be otherwise defined here. See implementation documentation for details of usage.

000D
When processing a session initiation RU, an SSCP has found two different resource definitions for the OLU, one with the real OLU name and one with the alias OLU name.

000E

When processing a session initiation RU, an SSCP has found two different resource definitions for the DLU, one with the real DLU name and one with the alias DLU name.

000F

The specified DLU network name is defined as a generic resource. The session should be re-initiated using the name of an LU.

0010

The LU 6.2 partner returned a name in the User Data field of its RSP(BIND) that differs from the name it returned in the User Data field of its RSP(BIND) for a previous BIND request. Either the partner changed its name or name changes in the network have caused delivery of the latest BIND request to a different partner.

0011

The LU 6.2 partner receiving a BIND request carrying one specific target secondary logical unit (SLU) name returned a name in the User Data field of its RSP(BIND) that is the same as it returned in response to a previous BIND carrying a different target SLU name. Name changes in the network name allowed two names to resolve to the same LU.

0012

The network qualifier of the name returned in the User Data field of a RSP(BIND) is not equal to the network identifier provided by the application that is using network qualified names. Name changes in the network have caused alteration of the network identifier.

0889xxxx Transaction Program Error

Explanation: The transaction program has detected an error, and issued a Send_Error call. This sense code is sent only in FMH-7.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

Occurs for either of the following reasons:

- Program Error—No Data Truncation. The transaction program sending data detected an error but did not truncate a logical record.
- Program Error—Purging. The transaction program receiving data detected an error. All remaining information, if any, that the receiving program had not yet received, and that the sending program had sent before being notified of the error, is discarded.

0001

Program Error—Data Truncation. The transaction program sending data detected an error and truncated the logical record it was sending.

0100
Occurs for either of the following reasons:

- Service Transaction Program Error—No Data Truncation. The service transaction program sending data detected an error and did not truncate a logical record.
- Service Transaction Program Error—Purging. The service transaction program receiving data detected an error. All remaining information, if any, that the receiving service transaction program had not yet received, and that the sending service transaction program had sent before being notified of the error, is discarded.

0101
Service Transaction Program Error—Data Truncation. The service transaction program sending data detected an error and truncated the logical record it was sending.

088B0000 BB Not Accepted—BIS Reply Requested

Explanation: Sent in response to a Begin Bracket Request (BB), either an LUSTAT bid or an Attach, to indicate that the receiver has sent a BIS request and wants to terminate the session without processing any more conversations, but without sending an UNBIND. A BIS reply is requested so that the negative response sender may send a normal UNBIND. This sense data is sent only by logical units (LUs) not supporting CNOS protocols.

088Cxxyy Missing Control Vector

Explanation: The RU or XID did not contain a required control vector.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

xxyy
Byte 2 contains the key (xx) of the required control vector that is missing. If more than one control vector is missing, only the first omission is reported.

0Eyy
Network Name (X'0E') control vector of type yy missing. Values for Network name type are:

X'F1'	PU name (not network qualified)
X'F3'	LU name
X'F4'	CP name
X'F5'	SSCP name
X'F6'	NNCP name
X'F7'	link station name (not network qualified)

1000
Product Set ID (X'10') control vector missing.

3200

Short-hold mode (X'32') control vector missing.

088F0000 XRF Procedure Error

Explanation: A request was received for an XRF-active or XRF-backup session and was not acted on.

0890xxxx Search Failure

Explanation: An error occurred while performing an APPN directory search for a resource.

0000

No specific code applies.

0010

Routing Error during a Directed Search. A Locate generalized data stream (GDS) variable for a directed search was received by an intermediate NNCP and could not be successfully routed to the destination control point.

0020

Resource Not Found during a Directed Search. A Locate generalized data stream (GDS) variable for a directed search was received by the named destination control point (CP) and the search argument resource is not a local resource.

0022

Destination of search not served by this control point (CP).

0024

A search request or BIND request was received from an unauthorized end node identifying an origin LU not represented in the APPN network node server's directory, and thus could not be authenticated.

0028

Resource Not Found, Broadcast Required. The resource was not found on a directed Locate search, and a restricted broadcast was executed at the destination and failed; a broadcast should be tried.

0030

Resource Deleted, No Broadcast Required. A Locate generalized data stream (GDS) variable for a directed search was received by the named destination control point (CP) and the search argument resource was deleted.

0038

Too Many Directed Search Subprocedures. A Locate search exceeded the maximum height of the search tree; too many directed search subprocedures were tried; no retry.

0040

Resource Not Found during a Broadcast Search. A Locate generalized data stream (GDS) variable for a broadcast search was received by a control point (CP) that does not provide network services for the search argument resource and neither do any of the CPs searched in its broadcast subtree. This condition is detected by crossing search requests (a CP sends and receives a search request with the same FQPCID and the same search argument resource) or by a local search failure and all CPs in the broadcast subtree returning this sense data.

0048

Neutral Reply Received from an APPN end node. A Locate reply with no Found and no Extended Sense Data (X'35') control vector was received from an APPN end node.

0050

Quiesced CP. A control point (CP) in the broadcast search tree is in a quiescent state and, therefore, not receiving Locate generalized data stream (GDS) variables. This condition is detected when a CP in the search subtree is quiesced and no other CP in the subtree found the requested resource.

0060

Storage Not Available. A control point (CP) in the broadcast search tree does not have sufficient storage to participate in the search and no other CP in the search subtree found the requested resource.

0070

Session Outage. A control point (CP) in the search tree has lost its CP-CP session with a CP that had been sent a Locate generalized data stream (GDS) variable and no reply had been received.

0080

Duplicate Fully-Qualified PCID. A control point (CP) in the search tree detected a duplicate fully-qualified PCID for a different session request from the session request that first used the fully-qualified PCID.

0891xxxx Invalid or Missing Invalid Network ID (NETID)

Explanation: The SNA Network ID (NETID) is invalid or missing.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

PLU NETID Invalid: The NETID of the primary logical unit (PLU) is not the same as that of the SSCP(PLU).

0002

Invalid NETID: The NETID field in CONNOUT does not match the NETID defined in the link station receiving the CONNOUT.

0003

Invalid NETID: The NETID field in the Request Network Address Assignment (RNAA) request unit is not the same as the native NETID. There is a mismatch between the system definitions of the SSCP and the type 4 node.

0004

Invalid Network ID (NETID) in the Network Name (X'0E', CP name) control vector. This sense data indicates that the Network Name control vector appended to the received XID3 does not contain a valid NETID. The NETID, preceding the control point (CP) name, must be from 1 to 8 bytes in length. This sense data may also indicate that the NETID has a valid length, but that the NETID is invalid for some other reason. For example, two adjacent APPN network nodes should not be configured with different NETIDs.

Note: When using the OS/2 Communications Manager, the NETID is also refered to as the “network name.”

0005

Invalid control point (CP) name in the Network Name (X'0E', CP name) control vector.

0895xxxx Control Vector Error

Explanation: The RU or XID contained a control vector that was in error.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

xxyy

The control vector with key xx at byte yy in XID3 is in error. Byte 2 (xx) contains the key of the control vector first detected in error. If more than one control vector is in error, only the first erroneous one is reported. Byte 3 (yy) of the sense-code-specific data contains the (0-origin) byte offset of the error within the control vector.

0897xxxx System Definition Mismatch

Explanation: The requested function is not supported by the receiver, or there is a mismatch between the sending and receiving system definitions.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

000A

The PU of an independent primary LU named in BFINIT does not have the same element address as the one in the adjacent link station (ALS) field of BFINIT. For example, when VTAM saw the BIND request (that is, NCP packaged it into a BFINIT), the PU2 element address that NCP told VTAM that the resource was located under was not found by VTAM. This could be a timing condition where the NCP has been taken down and reactivated.

This condition is referred to as “non-adjacent CP-CP sessions.” Many APPN implementations are dependent on the partner CP being adjacent. The requirement stems from CP-CP session activation having XID exchange dependencies. In order to support the proposed configuration, the local node would need to provide new function to support non-adjacent CP-CP sessions.

000D

Resource type not defined in receiver.

0011

APPN end node does not support locate. CP-CP sessions are being unbound because the adjacent node indicated that it does support CP-CP sessions but does not support receipt of LOCATE/CDINIT search requests.

0015

The origin LU (OLU) is represented using a dynamically defined resource, but the adjacent link station (ALS) selected to provide its services does not permit dynamic definitions. The condition is detected when a session initiation request is received for an independent LU and no predefinition is found for the OLU resource. The session initiation is rejected.

For example, this can occur when VTAM receives a session request from a workstation (such as the OS/2 Communications Manager), but the LU name of the originating LU (that is, your workstation) was not predefined to VTAM —and— the dynamic LU (DYNLU) parameter in VTAM specified that VTAM was not allowed to dynamically define the resource for you. If you want to use dynamic definition in VTAM, you need to get your VTAM system programmer to permit this.

0898xxxx Session Reset: The XRF Session is Being Reset

Explanation: The Extended Recovery Facility (XRF) session is being reset.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

The XRF-active session has been reset because the XRF-backup primary LU (PLU) forced a takeover.

0002

XRF-backup Hierarchical Reset: The identified XRF-active LU-LU session is being deactivated because the related XRF-active session terminated normally. The LU sending this sense data is resetting its half-session before receiving the response from the partner LU. (See UNBIND type X'12'.)

0003

XRF-active Hierarchical Reset: The identified XRF-active LU-LU session is being deactivated because the related XRF-backup session performed a forced takeover of this session (via SWITCH). The LU sending this sense data is resetting its half-session before receiving the response from the partner LU. (See UNBIND type X'13'.)

08A0xxxx Session Reset

Explanation: An LU or PU is resetting an LU-LU session.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

The LU is sending an UNBIND type X'0A' (SSCP gone); the identified LU-LU session had to be deactivated because of a forced deactivation of the associated SSCP-PU or SSCP-LU session, for example, because of a DACTPU, DACTLU, or DISCONTACT.

0002

The LU or SCM is sending UNBIND type X'0F' (cleanup).

0003

The gateway node is sending UNBIND type X'11' (gateway node cleanup); a gateway node is cleaning up the session because a gateway SSCP has directed the gateway node (via NOTIFY) to deactivate the session, for example, a session setup error or session takedown failure had occurred.

0004

Reversed FRSN values. The control point (CP) is sending an UNBIND type X'0F' (cleanup); the value in the Last FRSN Sent field is greater than the value in the Current FRSN field. (no retry)

0005

Topology Database Update (TDU) sent out of order. The control point (CP) is sending an UNBIND type X'0F' (cleanup); the value in the Last FRSN Sent field of the current TDU generalized data stream (GDS) variable is not equal to the value of the Current FRSN field in the TDU generalized data stream (GDS) variable that immediately preceded it. (no retry)

0006

Invalid FRSN Value. In CP Capabilities, the adjacent node indicated receipt of a Topology Database Update (TDU) with a FRSN value greater than the last one sent.

08A8xxxx Multiple-Domain Support Routing Exception

Explanation: The MDS router in the reporting network addressable unit (NAU) is unable to perform the required routing for an MDS_MU.

When this SNA report code is used in an SNA condition report (X'1532') generalized data stream (GDS) variable, the destination NAU name is included in the Reported on Location Name (X'09') subvector and the destination management services (MS) application name is included in the Reported On Agent (X'04') subvector of the condition report.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001

Destination NAU name unknown. Directory services could not locate the requested destination name.

0002

Directory services unavailable. No routing possible.

0003

Management services (MS) application name not recognized.

0004

Use of the CPSVCMG session not permitted. The reporting APPN network node has received an MDS_MU over a CPSVCMG session from another network node. These sessions are used for MDS_MUs only between an APPN network node and its served APPN end nodes.

0005

Function not supported by EN destination. The APPN end node destination does not support receipt of management services (MS) messages (reported by the serving APPN network node).

0006

Function not supported by APPN network node destination. The APPN network node destination does not support receipt of management services (MS) messages other than MS Capabilities and Alert (reported by the APPN network node performing routing).

0007

Function not supported by serving APPN network node. The serving network node of the APPN end node destination does not support routing of management services (MS) messages (reported by the APPN network node performing routing).

0008

Function not supported by EN. The reporting APPN end node has received an MDS_MU with a destination other than itself.

0009

Destination not supported by reporting APPN network node. An APPN network node has received an MDS_MU from another network node that cannot be routed. The destination is not the reporting network node itself nor is it one of the served APPN end nodes.

If the MDS_MU was routed based on unverified directory information (as indicated by the Routing verification indicator in the MDS Routing Information), the MDS_MU will be returned to the routing APPN network node along with the SNA condition report.

000A

Unrecoverable session failure. The MDS_SEND TP in the reporting node was unable to send the message because of an allocation error. Retries have been exhausted.

000B

Unrecoverable TP failure in remote node. The MDS_SEND TP in the reporting node was unable to send the message because of a TP failure in a remote node. Retries have been exhausted.

000C

Management services (MS) Application failure. The MDS router in the destination network addressable unit (NAU) is unable to communicate with the destination management services (MS) application program.

000D

Unrecoverable transaction program (TP) failure in reporting node. The MDS router in the reporting node was unable to send the message because of a local TP failure.

000E

Correlation error. An MDS_MU has been received that is not the first for a unit of work (First MDS Message indicator in the MDS Routing Information Message is OFF), but the unit of work correlator is unknown (does not match any active MDS transaction).

000F

Management services (MS) application congestion. The MDS router in the destination NAU is unable to communicate with the destination management services (MS) application because of local congestion (implementation buffer space for queuing additional MDS_MUs has been exhausted).

08A9xxxx Multiple-Domain Support Transaction Failure

Explanation: The reporting MDS router or management services (MS) application has detected a condition that has impacted an outstanding unit of work (identified by the unit-of-work correlator of the MDS Error message) or MDS_MU.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001
Failure caused by outage of a CPSVCMG session.

0002
Failure caused by outage of an SNASVCMG session. All retries have been exhausted.

0003
Unit of work canceled by reporting management services (MS) application program. The unit of work was canceled because of a timeout in the reporting management services (MS) application program.

0004
Unit of work canceled by reporting MDS Router. The unit of work was canceled by a garbage-collection timeout in the reporting MDS router.

0005
MDS router internal failure. The unit of work was canceled because of an internal failure in the reporting MDS router.

0006
MS Application internal error. The unit of work has been canceled either because the reporting MS application program was terminated or because another application program served by it was terminated. The type of program termination (normal or abnormal) is not indicated.

0007
MS Application router re-initialization. The unit of work has been canceled by the reporting MDS router because of a reinitialization of the application-level router.

08AAxxxx Required GDS Variable Missing

Explanation: The management services (MS) MDS_MU is missing a required generalized data stream (GDS) variable.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

xxxx
Bytes 2 and 3 following the sense code contain the ID of the missing GDS variable.

08B40000 CP-CP Sessions Not Required

Explanation: This sense data is sent from one APPN control point to another to deactivate CP-CP session between them, because the sender does not currently need CP-CP sessions, and the TG carrying the sessions is a limited resource.

This sense data is carried within the control vector (CV) X'35' portion of the UNBIND request (type X'01').

08B50000 Network Node Server Not Required

Explanation: This sense data is sent by an APPN end node control point (1) to deactivate CP-CP sessions with the NNCP, or (2) to reject a CP-CP session BIND request from the NNCP. The APPN end node no longer requires APPN network node services from the receiver.

This sense data is carried within the control vector (CV) X'35' portion of the UNBIND request (type X'01') for case (1) above, or on an UNBIND request (type X'FE') for case (2).

08B60000 CP-CP Sessions Not Supported

Explanation: This sense data is sent by an APPN network node control point to reject a CP-CP session BIND from another APPN control point; support for CP-CP sessions on that TG was removed since the time when the TG was first activated.

This sense data is carried within the control vector (CV) X'35' portion of the UNBIND request (type X'01').

X'10' Request Error

This category indicates that the RU was delivered to the intended network addressable unit (NAU) component, but could not be interpreted or processed. This condition represents a mismatch of NAU Capabilities.

1001xxxx RU Data Error

Explanation: Data in the request RU is not acceptable to the receiving component. For example, a character code is not in the set supported, a formatted data field is not acceptable to presentation services, or a value specified in the length field (LL) of a structured field is invalid.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0003

Isolated Pacing Message (IPM) Format Error. An incorrectly formatted IPM was received.

0024

A BIND request was received that was not for LU 6.2 and not in extended format. The BIND request is rejected.

0025

Unable to Extend BIND Request. An attempt made to add control vectors to a BIND request would have exceeded the maximum BIND length. The BIND is rejected.

1002xxxx RU Length Error

Explanation: The request RU was too long or too short.

0000

Indicates that no specific code applies.

1003xxxx Function Not Supported

Explanation: The function requested is not supported. The function may have been specified by a formatted request code, a field in an RU, or a control character.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

000D

The function identified in the request is not supported by the processing application transaction program.

0014

Cryptography is not supported, but a non-zero length was specified for the cryptography key.

0021

Mismatch between the session initiation request type and the LU type (independent or dependent).

This can occur if a local LU (defined as an independent LU) attempts to connect to a partner LU that is incorrectly defined as a dependent LU in the subarea node. There are other causes, but in every case, the cause involves a “session initiation type mismatch.” Your LU used dependent LU session initiation protocols (ACTLU, INIT_SELF) and the partner was expecting independent LU protocols, or your LU used independent LU session initiation protocols (BIND) and the partner expected dependent LU protocols.

Here are some examples:

- If using IBM's OS/2 Communications Manager and CPI-C, your program may choose to use a dependent local LU, instead of the default LU (which is the CP

LU, an independent LU). To use an alternate local LU, set the “APPCLLU” environment variable to the name of the local LU before starting the CPI-C program.

- Question: I'm connecting APPN nodes to our existing SNA subarea. I'm using control point names instead of XID ID numbers to match workstations to VTAM PU definitions. After the link activates, VTAM rejects workstation BIND requests with sense data X'10030021'. What's wrong?

  Answer: In your VTAM PU definition, is the PU name the same as the control point name? Because a PU is a VTAM resource, its name cannot match other resource names, such as LU names or control point names. Versions of VTAM before V4R1 assume that a workstation PU is not an LU. If a PU tries to send a BIND, VTAM rejects it with sense data X'10030021'.

  To fix your problem, make sure the PU name is different from the control point name on your VTAM PU definition. If you are not using the dynamic LU (DYNLU) feature of VTAM V3R4, add an LU definition in which the LU name matches the control point name.

  Example: PU and LU definitions for a workstation attached to a token-ring:

```
LINK0001 PU       ADDR=13,
                  CPNAME=CPNM0001,
                  PUTYPE=2,
                  DSCNT=NO

CPNM0001 LU       LOCADDR=0, MODETAB=MODAPPN
```

0022

BIND Received by an APPN end node which is not destination. A BIND was received which contained an Route Selection control vector (RSCV) specifying a destination other than this node. As this node is configured as an APPN end node, the BIND is rejected.

10050000 Parameter Error

Explanation: A parameter modifying a control function is invalid, or outside the range allowed by the receiver.

10070000 Category Not Supported

Explanation: DFC, SC, NC, or FMD request was received by a half-session not supporting any requests in that category; or an NS request byte 0 was not set to a defined value, or byte 1 was not set to an NS category supported by the receiver.

1008xxxx Invalid FM Header

Explanation: The FM header was not understood or translatable by the receiver, or an FM header was expected but not present. For LU 6.2, this sense code is sent in FMH-7 or UNBIND.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

200E

Invalid Concatenation Indicator. The concatenation indicator is on, but concatenation is not allowed.

201D

FM Header and Associated Data Mismatch. The FM header indicated associated data would or would not follow (for example, FM header 7 followed by log data, or FM header 5 followed by program initialization parameters), but this indication was in error; or a previously received RU (for example, -RSP(X'0846')) implied that an FM header would follow, but none was received.

4001

Invalid FM Header Type for this LU. The type of the FM header is other than 5, 7, or 12.

6000

FM Header Length Not Correct. The value in the FM header Length field differs from the sum of the lengths of the subfields of the FM header.

6005

Access Security Information Length Field Not Correct. The value in the Access Security Information Length field differs from the sum of the lengths of the Access Security Information subfields.

6009

Invalid Parameter Length. The field that specifies the length of fixed-length parameters has an invalid setting.

600B

Unrecognized FM Header Command Code. The partner LU received an FM header command code that it does not recognize. For LU 6.2, this sense data is sent only in FMH-7.

6011

Invalid Logical Unit of Work (LUW). The LUW Length field (in a Compare States generalized data stream (GDS) variable or an FMH-5) is incorrect, or the length field is invalid, or an LUW ID is not present but is required by the setting of the synchronization level field.

6021

Transaction Program Name Not Recognized. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a transaction program name that the receiver does not recognize. This sense data is sent only in FMH-7.

The partner LU rejected the incoming Attach because the local transaction program specified a TP name that the partner LU does not recognize.

This sense data can also indicate that the partner LU recognized the TP name, but could not start the program. One reason for this may be authorization problems. Some implementations of APPC (such as VM) check that three things match up: LU name, mode name, and TP name. These computers check the incoming user_id and password of each specific TP, and reject the Attach with this sense data if your program is not authorized to start the corresponding program on the remote computer.

Examples:

- If the remote location is using the OS/2 Communications Manager, its TP definition may have the TP name spelled wrong or be using the wrong combination of uppercase and lowercase characters. The TP name configured at the remote location must exactly match the tp_name specified on the Allocate call in your local program.

  The TP name field in the TP definition is one of the few APPC configuration fields that is case-sensitive. Be sure that the combination of uppercase and lowercase letters matches those specified in the program.

- If the remote location is running the VM operating system, this sense data will be received if the remote VM AVS is not authorized via the *IDENT control statements in VM to send information to the TP Program. This sense data is saying that there is no way to get to the TP Program on VM from the local node.

Programmer Response: Check the validity of the TP name, and the designated partner LU and mode names.

Operator Response: On the remote computer, check the list of TP names to be recognized. Ensure that they match the values supplied for the tp_name values on the Allocate call in the local computer.

If this checks out (that is, the partner TP is correctly defined on the remote computer), make sure the remote TP is properly authorized for the user_id and password sent on the Attach.

6031

PIP Not Allowed. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies program initialization parameter (PIP) data, but the receiver does not support PIP data for the specified transaction program. This sense data is sent only in FMH-7.

Explanation: The partner LU rejected the Attach because the local transaction program specified program initialization parameters (PIP data) and either the partner LU does not support PIP data or the partner transaction program has no PIP variables defined.

Programmer Response: Do not use PIP data when communicating with this remote transaction program. For example, APPC on the OS/2 Communications Manager does not accept incoming PIP data.

6032

PIP Not Specified Correctly. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a transaction program name that requires program initialization parameter (PIP) data, and either the FMH-5 specifies PIP data is not present or the number of PIP subfields present does not agree with the number required for the program. This sense data is sent only in FMH-7.

Explanation: The partner LU rejected the incoming Attach because the partner transaction program has one or more PIP variables defined and either the local transaction program has specified that no PIP variables are to be used, or the number of PIP variables defined by the local transaction program is different from the number specified by the remote transaction program.

Programmer Response: Specify that PIP data is to be used in the Allocate call, and make sure that the number of PIP variables agrees with the number required by the remote transaction program.

6034

Conversation Type Mismatch. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a conversation type that the receiver does not support for the specified transaction program. This sense data is sent only in FMH-7.

Explanation: The partner LU rejected the incoming Attach because it or the partner transaction program does not support the specified conversation type.

Programmer Response: Change the transaction program so it uses the proper conversation type (basic or mapped).

Operator Response: Alternatively, have the TP definition partner changed to reflect the conversation type used by the program.

6040

Invalid Attach Parameter. An Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a parameter that conflicts with the statement of LU capability previously provided in the BIND negotiation.

6041

Synchronization Level Not Supported. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies a synchronization level that the receiver does not support for the specified transaction program. This sense data is sent only in FMH-7.

Explanation: The partner LU rejected the incoming Attach because the local transaction program specified an unacceptable sync_level parameter. For example, the local transaction program issued an Allocate call with sync_level(CONFIRM), but at the remote computer it was configured as sync_level(NONE).

Programmer Response: Change the sync_level parameter on the Allocate call.

Operator Response: Alternatively, have the TP definition at the partner changed to reflect the sync_level used by the local transaction program.

6042

Reconnection Not Supported. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies reconnection support but the receiver does not support reconnection for the specified transaction program. This sense data is sent only in FMH-7.

6043

Unable to Reconnect Transaction Program—No Retry. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies the conversation correlator of a transaction program to which the receiver cannot reconnect. The condition is not temporary. This sense data is sent only in FMH-7.

6044

Unable to Reconnect Transaction Program—Retry Allowed. The Attach request (FMH-5) — sent because of an Allocate by the local program — specifies the conversation correlator of a transaction program to which the receiver cannot reconnect. The condition is temporary. This sense data is sent only in FMH-7.

1010xxxx Error on Locate Search or CP Capabilities Message Detected

Explanation: An error was detected during an APPN directory search or CP Capabilities exchange processing.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

Unrecoverable error, such as a duplicate control vector, was detected. This can be caused when another computer is using your control point LU name (CP LU), and has registered it with your network node (NN) server.

0001

A broadcast search resulted in two or more conflicting positive replies that differ on the control point (CP) owning the target resource. Multiple positive replies are acceptable, as long as all indicate the same owning CP.

0004

Unrecoverable error on CP Capabilities GDS variable exchange prevented its initiation or completion on a contention-winner CP-CP session.

1000

Length error in CP Capabilities generalized data stream (GDS) variable.

1002

Identifier error in CP Capabilities GDS variable.

4004

Incomplete negative or neutral reply received on a search, or reservation indicated on Broadcast, or “All” specified on a directed search.

5000

Length error in CD-Initiate GDS variable.

5002

No CD-Initiate (CDINIT) generalized data stream (GDS) variable returned on a search request.

5006

Session polarity or initiate type value received in CD-Initiate (CDINIT) generalized data stream (GDS) variable not supported.

500A

Mode name length error in CD-Initiate GDS variable.

A002

Find generalized data stream (GDS) variable not present on Locate search request.

B080

Command Parameters (X'80') control vector not present on Found generalized data stream (GDS) variable.

1014xxxx Control Vector Error on a Directory Services GDS Variable

Explanation: An error was detected on one of the following generalized data stream (GDS) variables: Locate, Find, Found, CDINIT, Register, or Delete.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

003C

Missing Associated Resource Entry (X'3C') control vector.

003D

Missing Directory Entry (X'3D') control vector.

0080

Invalid control vector.

023C

Conflicting APPN directory entry or invalid Associated Resource Entry (X'3C') control vector.

502B

No Route Selection control vector (RSCV) was received from an APPN network node server.

502C

No COS/TPF control vector received from an APPN network node server.

5046

TG vectors not present in a CD-Initiate (CDINIT) GDS variable from an APPN end node OLU or DLU.

A082

Missing Search Argument Directory Entry (X'82') control vector on Find.

B280

A Found from an APPN end node indicated the directory entry for a located resource was a wildcard entry.

1015xxxx XID Length Error

Explanation: The XID3 was too long or too short.

0001

Too few bytes in XID3.

0002

The length specified in the Length field in the XID3 is different from the number of bytes received. There is a mismatch between the number of bytes specified in the Length field of XID3 and the actual length of the received XID3.

1016xxxx XID Format 3 Parameter Error

Explanation: Data in the XID3 is not acceptable to the receiving component because the value in the received XID3 field, whose byte and bit offset is specified by the XID Negotiation Error (X'22') control vector (which also carries this sense code), is inconsistent with the corresponding field in the sent XID3.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001

The field in the received XID3 that specifies the Maximum number of I-frames that the sender can receive before acknowledgment is set to zero.

0002

The adjacent node has been inconsistent in its request for ACTPU. In a nonactivation XID3 exchange, it has changed the value of the ACTPU Suppression indicator sent in the previous XID3 exchange.

0003

The field in the received XID3 that specifies the maximum basic transmission unit (BTU) length that the sender can receive is set to less than 99 bytes, which is the minimum required.

0004

The received XID was not XID format 3 when XID format 3 was expected.

0005

The adjacent node does not support BIND segment generation but does support receipt of BIND segments. Any Type 2.1 node supporting receipt of BIND segments must also support generation of BIND segments.

0006

The adjacent node is an APPN end node, does not support BIND segment receipt, and has a maximum basic transmission unit (BTU) size of less than 265, the minimum required here.

Here's an anecdote: "I ran and formatted an OS/2 Communications Manager trace at the OS/2 network node and discovered that the NN was receiving a Sense data `X'10160006'`. I looked it up, and found out that the Macintosh SNA*ps 5250 was using a maximum BTU size of less than 265 (the minimum) and that the XID of the partner should be the NN instead of the AS/400. I increased the size and changed the XID and everything works fine now."

0007

The adjacent node is an APPN network node, does not support BIND segment receipt, and has a maximum basic transmission unit (BTU) size of less than 521, the minimum size required here.

0008

The adjacent node has changed from an APPN network node to an end node, or vice versa.

0009

The adjacent node is an APPN network node, does not provide control point (CP) services, and supports CP-CP sessions on this TG, a combination not allowed.

000A

During a nonactivation XID exchange, the adjacent node has changed the TG number that was negotiated during the link activation exchange.

000B

The adjacent node is the TG number negotiation winner and designates a TG number that the receiving node cannot allocate to this connection. When parallel TGs are supported between the two nodes, 0 is always such a number.

000C

The adjacent node is an APPN network node that does not support BIND segment generation, and this node's maximum basic transmission unit (BTU) size is less than 521.

Operator response: Increase the node BTU size to at least 521.

000D

The adjacent node is does not support the SDLC command response profile.

Operator response: The only defined value for this profile is X'0'. Make sure that the platform correctly sets that value (bits 4-7 of byte 12 of the XID).

000E

Two different Product Set ID (X'10') control vectors have been received from the adjacent node.

Operator response: There can be only one provided on the XID. Verify that the platform specifies only one Product Set ID on the XID.

000F

The link station roles specified in the sent and received negotiation-proceeding XID3s are not compatible. To activate a connection, one node must contain a primary link station; the other, a secondary link station.

0010

The support of combined asynchronous balanced mode (ABM) link stations indicated in the sent and received negotiation proceeding XID3s is not in agreement.

0011

This can occur for one of the following reasons:

- The adjacent node is attempting to activate parallel TGs, when parallel TGs are not supported between these two nodes.
- The link cannot be activated because the adjacent node does not support parallel TGs, and another link is active (or pending-active) to that node.

0012

(Retired)

The adjacent node has sent the Network Name (X'0E', CP name) control vector, but does not support the Exchange State indicators.

0013

The DLC Type field in the sent and received XID3s are not the same.

0014

The adjacent NRM link station sent an XID3, indicating that it contains a negotiable link station, after having indicated that it contained a link station with a nonnegotiable role.

0015

The adjacent node supports adaptive BIND pacing as a sender, but not as a receiver.

0016

The local node is attempting to activate a TG with a predefined number, but the adjacent node has attempted to use a non-zero number that is different from the one the local node used to represent the TG.

0017

After two negotiation proceeding XID exchanges, two negotiable link stations still have equal values in the Node Identification fields of their XID3s.

0018

The adjacent node is an APPN node but does not support adaptive BIND pacing as a sender and receiver.

001A

On two different TGs, the adjacent node has been inconsistent in its support of parallel TGs.

001B

The adjacent node provides or requests CP services, but does not support CP-CP sessions on this TG.

001C

The adjacent node sent a reserved value in the XID3 field defining its link station role.

001D

The adjacent node has only two-way alternating transmit-receive capability, but the local node supports only two-way simultaneous.

001E

The adjacent node does not include the Network Node (X'0E', CP name) control vector in its XID3, but indicates support of CP-CP sessions.

0020

The value sent in the Type of the XID Sending Node in XID3 indicates a node type with which the receiving node cannot activate a TG, for example, Type 2.1 nodes cannot activate TGs with Type 1 nodes.

101A0000 Invalid Control Vector Sequence

Explanation: A control vector was found containing a key that was invalid for the position of the control vector within a Topology Database Update (TDU).

X'20' State Error

This category indicates a sequence number error, or an RH or RU that it is not allowed for the receiver's current session control or data flow control state. These errors prevent delivery of the request to the intended component.

20010000 Sequence Number

Explanation: Sequence number received on normal-flow request was not 1 greater than the last.

20020000 Chaining

Explanation: Error in the sequence of the chain indicator settings (BCI, ECI), such as first, middle, first.

20030000 Bracket

Explanation: Error resulting from failure of sender to enforce bracket rules for session. (This error does not apply to contention or race conditions.)

20040000 Direction

Explanation: Error resulting from a normal-flow request received while the half-duplex flip-flop state was not Receive.

20080000 No Begin Bracket

Explanation: An FMD request specifying BBI=BB was received after the receiver had previously received a BRACKET INITIATION STOPPED request.

200A0000 Immediate Request Mode Error

Explanation: The immediate request mode protocol has been violated by the request.

200E0000 Response Correlation Error

Explanation: A response was received that cannot be correlated to a previously sent request.

20100000 BIS Protocol Error

Explanation: A BIS protocol error was detected; for example, a BIS request was received after a previous BIS was received and processed.

2011xxxx Pacing Protocol Error

Explanation: A violation of pacing protocols was detected.

0000

A normal-flow or BIND request was received after the pacing count had been reduced to zero and before a pacing response had been sent.

0001

Unexpected Isolated Pacing Message (IPM) Received. An IPM was received when the receiver was in a state that did not allow it.

0002

Unexpected Pacing Request Received. A request with the pacing indicator set was received when the receiver was in a state that did not allow it.

0003

Pacing Response Indicator incorrectly set. The pacing indicator was set in a non-IPM response received while adaptive pacing was being used.

20120000 Invalid Sense Code Received

Explanation: A negative response was received that contains an SNA-defined sense code that cannot be used for the sent request.

X'40' RH Usage Error

This category indicates that the value of a field or combination of fields in the RH violates architectural rules or previously selected BIND options. These errors prevent delivery of the request to the intended component and are independent of the current states of the session. They may result from the failure of the sender to enforce session rules. Detection by the receiver of each of these errors is optional.

40030000 BB Not Allowed

Explanation: The Begin Bracket indicator (BBI) was specified incorrectly; for example, BBI=BB with BCI=¬BC.

40070000 Definite Response Not Allowed

Explanation: Definite response was requested when not permitted.

40110000 Incorrect Specification of RU Category

Explanation: The RU Category indicator was specified incorrectly; for example, an expedited-flow request or response was specified with RU Category indicator = FMD.

40120000 Incorrect Specification of Request Code

Explanation: The request code on a response does not match the request code on its corresponding request.

40130000 Incorrect Specification of (SDI, RTI)

Explanation: The Sense Data Included indicator (SDI) and the Response Type indicator (RTI) were not specified properly on a response. The proper value pairs are (SDI=SD, RTI=negative) and (SDI=¬SD, RTI=positive).

40140000 Incorrect Use of (DR1I, DR2I, ERI)

Explanation: The Definite Response 1 indicator (DR1I), Definite Response 2 indicator (DR2I), and Exception Response indicator (ERI) were specified incorrectly; for example, a SIGNAL request was not specified with DR1I=DR1, DR2I=¬DR2, and ERI=¬ER.

40150000 Incorrect Use of QRI

Explanation: The Queued Response indicator (QRI) was specified incorrectly; for example, QRI=QR on an expedited-flow request.

40190000 Incorrect Indicators with Last-In-Chain Request

Explanation: A last-in-chain request has specified incompatible RH settings; for example, RQE*, CEBI=¬CEB, and CDI=¬CD.

X'80' Path Error

This category indicates that the request could not be delivered to the intended receiver, because of a path outage, an invalid sequence of activation requests, or one of the listed path information unit (PIU) errors. Some PIU errors fall into other categories; for example, sequence number errors are sense code category `X'20'`. A path error received while the session is active generally indicates that the path to the session partner no longer exists.

80010000 Intermediate Node Failure

Explanation: There has been been a failure, such as a hardware failure or program check, in a node providing intermediate routing function. A response may or may not be possible.

80020000 Link Failure

Explanation: Data link failure. It is possible that the APPC subsystem is not active on the partner computer.

8003xxxx NAU Inoperative

Explanation: The network addressable unit (NAU) is unable to process requests or responses; for example, the NAU has been disrupted by an abnormal termination.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001
Hierarchical Reset. The identified LU-LU session is being deactivated; an ACTLU/ACTPU(Cold) or DACTLU/DACTPU was received, or the PU has failed.

0003

Unrecoverable LU Failure. The identified LU-LU session had to be deactivated because of an abnormal termination of the PLU or SLU; recovery from the failure was not possible.

0004

Recoverable LU Failure. The identified LU-LU session had to be deactivated because of an abnormal termination of one of the LUs of the session; recovery from the failure may be possible.

8004xxxx Unrecognized Destination

Explanation: A node in the path has no routing information for the destination specified either by the secondary logical unit (SLU) name in a BIND request or by the transmission header (TH).

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

The partner LU name was not recognized. This can occur if you did not have the LU name specified in the BIND secondary logical unit (SLU) name field defined. For example, your program issues an Allocate call that uses a fully-qualified partner LU name, but includes the partner LU name only--without the NETID.

Note: When using the OS/2 Communications Manager, the NETID is also referred to as the “network name.”

0001

A request was received by a gateway function that could not be rerouted because of invalid or incomplete routing information.

8005xxxx No Session

Explanation: No half-session is active in the receiving APPN end node for the indicated origination-destination pair, or no boundary function session connector is active for the origin-destination pair in a node providing the boundary function. A session activation request is needed.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

The receiver received a request other than a session control request when no LU-LU session was active.

0002
The receiver received a request other than a session control request when no LU-SSCP session was active.

0003
The receiver received a session control request other than BIND/UNBIND when no LU-LU session was active.

0004
The receiver received an UNBIND when no LU-LU session was active.

0005
The receiver received a session control request other than ACTLU/DACTLU for the LU-SSCP session when no LU-SSCP session was active.

0006
The receiver received DACTLU when no LU-SSCP session was active.

8007xxxx Segmenting Error

Explanation: First BIU segment had less than 10 bytes; or Mapping field sequencing error, such as first, last, middle; or segmenting not supported and Mapping field not set to BBIU, EBIU.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000
No specific code applies.

0001
The node does not support receipt of segments, and a Mapping field value other than BBIU, EBIU was received. Sent in UNBIND.

0002
Interleaved BIND Segments Not Allowed: A BIND receiver that is in the middle of receiving segments of one BIND receives a segment from a different BIND; the receiver rejects both BINDs and disconnects the link.

80080000 PU Not Active

Explanation: The SSCP-PU secondary half-session in the receiving node has not been activated and the request was not ACTPU for this half-session. For example, the request was ACTLU from an SSCP that does not have an active SSCP-PU session with the PU associated with the addressed LU.

80090000 LU Not Active

Explanation: The destination address specifies an LU for which the SSCP-LU secondary half-session has not been activated, and the request was not ACTLU.

800A0000 Too-long PIU

Explanation: Transmission was truncated by a receiving node because the path information unit (PIU) exceeded a maximum length or sufficient buffering was not available.

800B0000 Incomplete TH

Explanation: Transmission received was shorter than a transmission header (TH).

800C0000 DCF Error

Explanation: Data Count Field (DCF) was inconsistent with transmission length.

800D0000 Lost Contact

Explanation: Contact with the link station for which the transmission was intended has been lost, but the link has not failed. If the difference between link failure and loss of contact is not detectable, link failure (X'8002') is sent.

800E0000 Unrecognized Origin

Explanation: The origin address specified in the transmission header (TH) was not recognized.

800Fxxxx Invalid Address Combination

Explanation: The address combination is invalid.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

The (DAF', OAF') (FID2) combination or the LSID (FID3) specified an invalid type of session, for example, a PU-LU combination.

For example, this sense data may be seen when a computer attempts to set up a session (by sending a BIND) to a version of VTAM (probably a version before 3.2) that does not support independent sessions on the link being used. This may be observed as a single SNASVCMG session being established, but not the usual pair of them.

0001

The FID2 ODAI setting in a received BIND is incorrect; the BIND is rejected.

80100000 Segmented RU Length Error

Explanation: An RU was found to exceed a maximum length, or required buffer allocation that might cause future buffer depletion.

8014xxxx No Path Exists to the Destination Node

Explanation: Route Selection services in the control point (CP) has determined from the APPN topology database that no path exists to the destination node.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0001

No route to the destination node exists for the specified class of service (COS). This can occur when using one of the modes that specifies a particular COS, such as #BATCHSC, that cannot be achieved along any route through the network. The implication is that a path does exist, but not a path with the required characteristics.

With the OS/2 Communications Manager, for example, examine whether the DEFINE_PARTNER_LU_LOCATION fq_owning_cp_name for the host LU matches the adjacent_node_cp_name returned on your display of the active link. If it does not, the NN cannot find a route to the CP name you specified on the DEFINE_PARTNER_LU_LOCATION.

0002

Invalid class-of-service (COS) name received.

0003

The APPN topology database indicates that the destination node is not available now; the node either has inconsistent data or is quiescing.

0004

The APPN topology database indicates that the endpoint resources are depleted; the node is out of either half-session control blocks or message buffers.

0005

The Route Selection control vector (RSCV) specifying the route to the destination node was truncated. The length of the generated RSCV exceeds the maximum allowed.

8020xxxx Session Reset

Explanation: The LU-LU session identified in the UNBIND is being deactivated because of a reset condition.

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0000

No specific code applies.

0001

Virtual Route Inoperative. The virtual route used by the LU-LU session has become inoperative, thus forcing the deactivation of the identified LU-LU session.

0003

Virtual Route Deactivated. The identified LU-LU session had to be deactivated because of a forced deactivation of the virtual route being used by the LU-LU session.

0004

Route Extension Failure. The route extension used by the LU-LU session has become inoperative, thus forcing the deactivation of the identified LU-LU session.

0007

Third Party Termination. The network operator cause the forced or cleanup termination of the LU-LU session.

X'FF' Product Unique Errors

This category indicates a condition was detected by the node that is not defined in SNA. The sense code defined is unique to a platform implementation.

FFFExxxx OS/2 Communications Manager: Unable to Obtain Session

Bytes 2 and 3 following the sense code contain sense-code-specific information:

0011

CNOS Negotiation in Progress. An Allocate call was attempted while CNOS contention winner negotiation is occurring in another process or thread, from the same local LU to the same partner LU. Session limits are currently being negotiated with your partner, because of a previous allocation attempt or a CNOS verb. This condition will continue until the previous session limit negotiation is completed.

Programmer Response: Retry the operation. This condition is timing related.

0012

APPC Not Started. An Allocate call attempted while APPC has not been started.

Programmer Response: Retry the operation. This condition is timing related.

0013

A mode, other than CPSVCMG or SNASVCMG, is pending a delete. A program is unable to issue an Allocate call for this mode while the mode is in pending delete state. With the OS/2 Communications Manager, pending delete state is attained by issuing DEFINE_MODE with changes, or DELETE_MODE. These verbs do not go into effect until all sessions using this mode are inactive.

0014

The CPSVCMG mode is pending a delete. APPC is unable to internally issue an Allocate call for this mode while the mode is in pending delete state.

0015

The SNASVCMG mode is pending a delete. APPC is unable to internally issue an Allocate call for this mode while the mode is in pending delete state. Pending delete state is attained by issuing DEFINE_MODE with changes, or DELETE_MODE. These verbs do not take affect until all sessions using this mode are inactive.

0016

Session unavailable on a mode other than CPSVCMG or SNASVCMG. An Allocate call with return_control(WHEN_SESSION_FREE) was attempted, but no pending or bindable sessions are available.

0017

A session is unavailable on a CPSVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_SESSION_FREE), but no pending or bindable sessions are currently available.

0018

A session is unavailable on a SNASVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_SESSION_FREE), but no pending or bindable sessions are currently available.

0019

Partner CNOS race condition. APPC issued an internal CNOS verb. Although its primary return was PARTNER_REJECT and the secondary return code was CNOS_RACE, no sense data associated with the returned verb was generated.

Programmer Response: Retry the operation. This condition is timing related.

001A

This code is returned when a second ACTLU or a DACTPU is received on a session that is pending activation. This occurs only during sub-area session activation.

0101

Last session deactivated on a mode other than CPSVCMG or SNASVCMG. This code is returned on an Allocate call when return_control is WHEN_SESSION_ALLOCATED, or WHEN_SESSION_FREE. A queued session request is rejected because the last session on the requested mode was deactivated. Therefore, the Allocate call returns without being fulfilled, with this sense data.

0102

Last session deactivated on a CPSVCMG Mode. APPC issued an internal Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE). A queued session request is rejected because the last session on the requested mode was deactivated. Therefore the Allocate call is returned without being fulfilled, with this sense data.

0103

Last session deactivated on an SNASVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE). A queued session request is rejected because the last session on the requested mode was deactivated. Therefore the Allocate call is returned without being fulfilled, with this sense data.

0104

Last contention winner session is deactivated on a mode other than CPSVCMG or SNASVCMG. This code is returned on an Allocate call when return_control(WHEN_CONWINNER_ALLOCATED). A queued session request is rejected because the last session on the requested mode was deactivated. Therefore, the Allocate call returns without being fulfilled, with this sense data.

0105

Last contention winner session is deactivated on a CPSVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_CONWINNER_ALLOCATED). A queued session request is rejected because the last session on the requested mode was deactivated. Therefore the Allocate call is returned without being fulfilled, with this sense data.

0106

Last contention winner session is deactivated on an SNASVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_CONWINNER_ALLOCATED). A queued session request is rejected because the last session on the requested mode was deactivated. Therefore the Allocate call is returned without being fulfilled, with this sense data.

0107

Associated session for a conversation group is deactivated. This code is returned on an Allocate call when return_control(WHEN_CONV_GROUP_ALLOC). A queued session request is rejected because the last session on the requested mode was deactivated. Therefore, the Allocate call returns without being fulfilled, with this sense data.

0108

Associated Session for Conversation Group is Not Defined. An Allocate call was issued, with return_control(WHEN_CONV_GROUP_ALLOC) was specified, and no session is available. A likely cause of this sense data is that your program supplied an invalid conversation_group_id parameter on an Allocate call. In OS/2 Communications Manager, a list of valid conversion group IDs can be found by issuing the DISPLAY verb, requesting Active TP Information.

0109

Contention winner session never available on a mode other than CPSVCMG or SNASVCMG. An Allocate call with return_control(WHEN_CONWINNER_ALLOCATED) was specified and the mode's minimum contention winner value is zero and no session is available.

010A

Contention winner session never available on CPSVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_CONWINNER_ALLOCATED) and the mode's minimum contention winner value is zero and no session is available.

010B

Contention winner session never available on SNASVCMG mode. APPC issued an internal Allocate call with return_control(WHEN_CONWINNER_ALLOCATED) and the mode's minimum contention winner value is zero and no session is available.

010C

Session never available on a mode other than CPSVCMG or SNASVCMG. This sense data can be caused by one of the following conditions:

- An Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE) was issued. The mode's session limit is zero and no session is available.
- An Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE) was issued. The target computer has set the mode session limit for this mode to zero.
- The BIND for a subarea session was successful, but APPC changed the number of sessions to zero before ACTLU arrived.

010D

Session never available on a CPSVCMG mode. This sense data can be caused by one of the following conditions:

- APPC issued an internal Allocate call verb with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE). The mode's session limit is zero and no session is available.
- APPC issued an internal Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE). The target computer has set the mode session limit for this mode to zero.
- The BIND for a subarea session was successful, but APPC changed the number of sessions to zero before ACTLU arrived.

010E

Session never available on an SNASVCMG mode. This sense data can be caused by one of the following conditions:

- APPC issued an internal Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE). The mode's session limit is zero and no session is available.

- APPC issued an internal Allocate call with return_control(WHEN_SESSION_ALLOCATED) or return_control(WHEN_SESSION_FREE). The target computer has set the mode session limit for this mode to zero.
- The BIND for a subarea session was successful, but APPC changed the number of sessions to zero before ACTLU arrived.

010F

The SNASVCMG mode does not exist. If the SNASVCMG mode does not exist, and APPC is executing its internal CNOS transaction program to control parallel sessions, this sense data will occur. Since the SNASVCMG mode should always be defined, this signals an error condition.

0110

A remote mode, other than CPSVCMG or SNASVCMG, is closed or does not exist. Usually, this means that the mode is not defined correctly at the partner, or the number of sessions was changed to zero.

APPC issued an internal CNOS verb, and its primary return was PARTNER_REJECT and the secondary return code was MODE_CLOSED or BAD_MODE, and there was no sense data.

0111

A remote CPSVCMG mode is closed or does not exist. APPC issued an internal CNOS verb. Its primary return was PARTNER_REJECT and the secondary return code was MODE_CLOSED or BAD_MODE. There was no sense data associated with the returned verb.

Usually, this means that the mode is not defined correctly at the partner, or the number of sessions was changed to zero.

0112

A remote SNASVCMG mode is closed or does not exist. APPC issued an internal CNOS verb, and its primary return was PARTNER_REJECT and the secondary return code was MODE_CLOSED or BAD_MODE, and there was no sense data associated with the returned verb.

Usually this means that the mode is not defined correctly at the partner, or the number of sessions was changed to zero.

0113

Link is defined as independent LUs only. The network addressable unit (NAU) was configured as zero; this link can only be used for independent logical units (LUs). The primary cause of this is that the PU statement in the configuration has aparameter XID=NO; this must be YES if the host is going to allow OS/2 to send independent LU BINDs.

0114

Path Control ID is Mismatched for Subarea. Mismatch between the PC IDs in the ACTLU and the INITSELF RUs.

0115

The CNOS TP on the target node provided an invalid response. This sense data indicates an internal SNA error on the target node.

This sense data may be returned (in error) if your program is attempting to BIND non-CP sessions with a VTAM control point LU. VTAM may reject the BIND request with this sense data.

FFFFxxxx OS/2 Communications Manager: Directory Function Error

Explanation: An error occurred while performing an APPN directory search or directory delete.

Note: Products use the X'FFFF' prefix on sense codes that are product dependent and, more importantly, are never sent. They are only returned on an API crossing when the architected primary and secondary return codes do not provide adequate information and there is no architected sense code with the required definition.

0002

Multiple Resources Found; Resource Must be Unique. A network search for a resource required to have a unique name has returned multiple positive replies from different owners. Only one resource should have that name. The other resources should be renamed or deleted, otherwise the first resource found in the search will be returned.

0003

No link is available to an APPN network node server. No NNCP server was found when a local program issued an Allocate call to get a session with its associated partner LU.

This sense data indicates that a session activation failed because APPC could not find any link to connect it to the partner node. If your node is serviced by an APPN network node, this sense data means that your node could not contact that network node (probably because the link is down or defined incorrectly).

For example, when using the OS/2 Communications Manager in your local node, check to see that it is serviced by an APPN network node. If not, it means that you have not provided a valid DEFINE_PARTNER_LU_LOCATION together with a DEFINE_LOGICAL_LINK verb which specifies the link to use. Assure that the DEFINE_LOGICAL_LINK verb contains a valid fq_adjacent_cp_name parameter and the link is either active, or is defined as activate_at_startup(NO) (which brings it up right away, as opposed to when the OS/2 Communications Manager is started).

Summary: either

- Your node does not have a link to an APPN network node; thus it has no place to send the Locate RU.

- Your node does not have a link to the target node, where the searched-for LU is located.

Examples of how this can happen:

- The target MAC address for the APPN network node is wrong.
- The data link control (DLC) was deactivated on either computer.
- APPC is not fully started yet on your local computer, and the links have not yet been activated.
- The partner LU name is just wrong. The partner LU name is spelled wrong or does not exist. If the application is using CPI-C, check the CPI-C side information table to assure that the correct partner LU name is specified.

If you had this session working before, it is likely that your link to your partner or network node has gone down. Thus you could have failures now even though you have not changed your configuration. If the network node goes down and comes back up, your link is not automatically re-established. (Look for the NNLINK package for a tool to keep that link up).

To check this out, go to the Logical Link Services, and see if your link is displayed as inactive. If it is, re-activate it using the same panel. (If it's not, there's some other problem -- probably a configuration problem.)

0006

Cannot delete a local node directory entry. An APPN end node sent a Delete generalized data stream (GDS) variable, requesting that the CP LU be deleted. This cannot be performed.

0007

Address space unknown. The node has received data or is trying to send data on a link station which is not in its local address space. The node has canceled processing of the data.

Appendix A. Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the

IBM Director of Licensing,
IBM Corporation,
500 Columbus Avenue,
Thornwood, NY, 10594
USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Limited
Department 071
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM
ADSTAR
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
CICS
C Set++
C/370
DATABASE 2
DatagLANce
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
Distributed Relational Database Architecture
DRDA
Extended Services
FFST
First Failure Support Technology
IBM
IMS
Lan Distance
MVS/ESA
MVS/XA
NetView
OS/400
OS/390
OS/2
PowerPC
QMF
RACF
RISC System/6000
SAA
SP
SQL/DS
SQL/400
S/370
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WIN-OS/2

Trademarks of Other Companies

The following terms are trademarks or registered trademarks of the companies listed:

C-bus is a trademark of Corollary, Inc.

HP-UX is a trademark of Hewlett-Packard.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Solaris is a trademark of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Contacting IBM

This section lists ways you can get more information from IBM.

If you have a technical problem, please take the time to review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. Depending on the nature of your problem or concern, this guide will suggest information you can gather to help us to serve you better.

For information or to order any of the DB2 Universal Database products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

Telephone

If you live in the U.S.A., call one of the following numbers:

- 1-800-237-5511 to learn about available service options.
- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by selecting the "Roadmap to IBM Support" item at: http://www.ibm.com/support/.

Note that in some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

World Wide Web
http://www.software.ibm.com/data/
http://www.software.ibm.com/data/db2/library/

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more. The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information. (Note that this information may be in English only.)

Anonymous FTP Sites
ftp.software.ibm.com

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools concerning DB2 and many related products.

Internet Newsgroups
comp.databases.ibm-db2, bit.listserv.db2-l

These newsgroups are available for users to discuss their experiences with DB2 products.

CompuServe
GO IBMDB2 to access the IBM DB2 Family forums

All DB2 products are supported through these forums.

To find out about the IBM Professional Certification Program for DB2 Universal Database, go to http://www.software.ibm.com/data/db2/db2tech/db2cert.html

Part Number: B2CONNA

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.