z/OS Communications Server

SNA Programmer's LU 6.2 Reference

Version 1 Release 2

SC31-8810-00

z/OS Communications Server

SNA Programmer's LU 6.2 Reference

Version 1 Release 2

SC31-8810-00

Note:
Before using this information and the product it supports, be sure to read the general information under “Appendix B. Notices” on page 679.

First Edition (October 2001)

This edition applies to Version 1 Release 2 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

Publications are not stocked at the address given below. If you want more IBM® publications, ask your IBM representative or write to the IBM branch office serving your locality.

Address comments to:
IBM Corporation
Software Reengineering
Department G71A/ Bldg 503
Research Triangle Park, North Carolina 27709–9990
U.S.A.

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):
1-800-227-5088

Internet e-mail:
usib2hpd@vnet.ibm.com

World Wide Web:
http://www.ibm.com/servers/eserver/zseries/zos/

IBMLink™:
CIBMORCF at RALVM17

IBM Mail Exchange:
tkinlaw@us.ibm.com

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1988, 2001. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures vii

Tables ix

About This Book xi
Who Should Use This Book xi
Typographic Conventions Used in This Book xi
Where to Find More Information xi
Where to Find Related Information on the Internet xi
Licensed Documents xii
LookAt, an Online Message Help Facility xiii
How to Contact IBM Service xiii
z/OS Communications Server Information xiv

Summary of Changes xxi

Chapter 1. LU 6.2 Macroinstruction Syntax and Operands 1
About This Chapter 1
How Macroinstructions Are Described 1
Syntax Descriptions 1
Operand Descriptions 2
Completion Information 3
Coding Default Values. 3
Coding Comments 3
Coding Continued Lines 4
How to Read the Syntax Diagrams 4
APPCCMD CONTROL=ALLOC, QUALIFY=ALLOCD 7
APPCCMD CONTROL=ALLOC, QUALIFY=CONVGRP 19
APPCCMD CONTROL=ALLOC, QUALIFY=CONWIN. 31
APPCCMD CONTROL=ALLOC, QUALIFY=IMMED 43
APPCCMD CONTROL=ALLOC, QUALIFY=WHENFREE 54
APPCCMD CONTROL=CHECK 66
APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDPROG 68
APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDSERV 76
APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDTIME. 83
APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDUSER 90
APPCCMD CONTROL=DEALLOC, QUALIFY=CONFIRM 98
APPCCMD CONTROL=DEALLOC, QUALIFY=DATACON. 106
APPCCMD CONTROL=DEALLOC, QUALIFY=DATAFLU 118
APPCCMD CONTROL=DEALLOC, QUALIFY=FLUSH 129
APPCCMD CONTROL=DEALLOCQ 136
APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS 144
APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS 150
APPCCMD CONTROL=OPRCNTL, QUALIFY=DACTSESS 160
APPCCMD CONTROL=OPRCNTL, QUALIFY=DEFINE 165
APPCCMD CONTROL=OPRCNTL, QUALIFY=DISPLAY 172
APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE 179
APPCCMD CONTROL=PREALLOC, QUALIFY=ALLOCD. 186
APPCCMD CONTROL=PREALLOC, QUALIFY=CONVGRP 197
APPCCMD CONTROL=PREALLOC, QUALIFY=CONWIN 208
APPCCMD CONTROL=PREALLOC, QUALIFY=IMMED 220
APPCCMD CONTROL=PREALLOC, QUALIFY=WHENFREE 230
APPCCMD CONTROL=PREPRCV, QUALIFY=CONFIRM. 241

© Copyright IBM Corp. 1988, 2001

APPCCMD CONTROL=PREPRCV, QUALIFY=DATACON 250
APPCCMD CONTROL=PREPRCV, QUALIFY=DATAFLU 262
APPCCMD CONTROL=PREPRCV, QUALIFY=FLUSH 274
APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY 280
APPCCMD CONTROL=RCVEXPD, QUALIFY=IANY 288
APPCCMD CONTROL=RCVEXPD, QUALIFY=ISPEC 296
APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC 304
APPCCMD CONTROL=RCVFMH5, QUALIFY=DATAQUE 312
APPCCMD CONTROL=RCVFMH5, QUALIFY=NULL 325
APPCCMD CONTROL=RCVFMH5, QUALIFY=QUEUE 334
APPCCMD CONTROL=RECEIVE, QUALIFY=ANY 343
APPCCMD CONTROL=RECEIVE, QUALIFY=IANY 355
APPCCMD CONTROL=RECEIVE, QUALIFY=ISPEC 366
APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC 378
APPCCMD CONTROL=REJECT, QUALIFY=CONV 390
APPCCMD CONTROL=REJECT, QUALIFY=CONVGRP 396
APPCCMD CONTROL=REJECT, QUALIFY=SESSION. 402
APPCCMD CONTROL=RESETRCV 408
APPCCMD CONTROL=SEND, QUALIFY=CONFIRM 415
APPCCMD CONTROL=SEND, QUALIFY=CONFRMD 424
APPCCMD CONTROL=SEND, QUALIFY=DATA 431
APPCCMD CONTROL=SEND, QUALIFY=DATACON 444
APPCCMD CONTROL=SEND, QUALIFY=DATAFLU 456
APPCCMD CONTROL=SEND, QUALIFY=ERROR 468
APPCCMD CONTROL=SEND, QUALIFY=FLUSH 482
APPCCMD CONTROL=SEND, QUALIFY=RQSEND 490
APPCCMD CONTROL=SENDEXPD, QUALIFY=DATA 497
APPCCMD CONTROL=SENDFMH5, QUALIFY=NULL 506
APPCCMD CONTROL=SENDRCV, QUALIFY=DATAFLU 512
APPCCMD CONTROL=SETSESS, QUALIFY=RESUME 526
APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND 531
APPCCMD CONTROL=SETSESS, QUALIFY=SYNCBEG 536
APPCCMD CONTROL=SETSESS, QUALIFY=SYNCEND 541
APPCCMD CONTROL=TESTSTAT, QUALIFY=ALL 546
APPCCMD CONTROL=TESTSTAT, Qualifiable 552
APPCCMD CONTROL=TESTSTAT, QUALIFY=ISPEC 558
APPCCMD CONTROL=TESTSTAT, QUALIFY=SPEC 565
ISTGAPPC 572
ISTRPL6. 575

Chapter 2. Return Codes 585
RCPRI and RCSEC Codes 585
RTNCD, FDB2 Information for LU 6.2 623

Chapter 3. DSECTs 625
BIND Image (ISTDBIND). 625
FMH-5 (ISTFM5). 635
RPL Extension (ISTRPL6X). 637
CNOS Session Limits Data Structure (ISTSLCNS) 645
DEFINE/DISPLAY Session Limits Data Structure (ISTSLD) 646
Restore Data Structure (ISTSREST) 648
Status Data Structure (ISTSTATD) 648
Feedback Code Data Structure (ISTUSFBC) 649
APPCCMD Vector Lists (ISTAPCVL) 658
Application-ACB Vector List (ISTVACBV) 663
Access-Method-Support Vector List (ISTAMSVL) 665

Resource-Information Vector List (ISTRIVL) 669
Extended Buffer List Entry (ISTBLXEN) 673

Chapter 4. Summary of Register Usage 675

Appendix A. Information Apars 677
IP Information Apars 677
SNA Information Apars 678

Appendix B. Notices 679
Programming Interface Information 681
Trademarks. 682

Index 685

Figures

1. How to Code Comments and Continuation Lines 3
2. Layout of the RPL Extension 643

© Copyright IBM Corp. 1988, 2001

Tables

1. Format of WHATRCV Mask 376
2. LU 6.2 Global Macro Variables Set by ISTGAPPC 572
3. Register Contents upon Return of Control 675
4. IP Information Apars 677
5. SNA Information Apars 678

© Copyright IBM Corp. 1988, 2001

About This Book

This manual is designed to help customers write VTAM® application programs to use the VTAM logical unit (LU) 6.2 application programming interface (API). This manual describes the format of the macroinstructions and presents each macroinstruction in alphabetical order.

This manual explains macroinstruction syntax and parameters, return codes and responses, and identifies fields set by DSECTs. *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* also explains how to design VTAM LU 6.2 application programs.

Who Should Use This Book

This book is for programmers (such as application or system programmers) who code VTAM application programs. This audience can include programmers who are modifying existing programs or writing new ones.

You should be familiar with LU 6.2 architecture before you write LU 6.2 programs. *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* provides this familiarity and is, therefore, a corequisite for using the *z/OS Communications Server: SNA Programmer's LU 6.2 Reference*. You should also be familiar with information in *z/OS Communications Server: SNA Programming*.

You should also be familiar with the information in the assembler language documentation for your operating system.

Typographic Conventions Used in This Book

This publication uses the following typographic conventions:

- Commands that you enter verbatim onto the command line are presented in **bold**.
- Variable information and parameters that you enter within commands, such as filenames, are presented in *italic*.
- System responses are presented in `monospace`.

Note: In this manual, macroinstruction operands are separated by blanks. This convention is used to improve the formatting of the manual. When coding the macroinstructions, you must delete the blanks.

Where to Find More Information

This section contains:

- Pointers to information available on the Internet
- Information about licensed documentation
- Information about LookAt, the online message tool
- A set of tables that describes the books in the z/OS Communications Server (z/OS CS) library, along with related publications

Where to Find Related Information on the Internet

Home Page	Web address
z/OS	http://www.ibm.com/servers/eserver/zseries/zos/

© Copyright IBM Corp. 1988, 2001

z/OS Internet Library
http://www.ibm.com/servers/eserver/zseries/zos/bkserv/
IBM Communications Server product
http://www.software.ibm.com/network/commserver/
IBM Communications Server support
http://www.software.ibm.com/network/commserver/support/
IBM Systems Center publications
http://www.redbooks.ibm.com/
IBM Systems Center flashes
http://www-1.ibm.com/support/techdocs/atsmastr.nsf
VTAM and TCP/IP
http://www.software.ibm.com/network/commserver/about/csos390.html
IBM http://www.ibm.com
RFC http://www.ietf.org/rfc.html

Information about Web addresses can also be found in informational APAR II11334.

DNS Web Sites

For information about DNS, see the following Web sites:

USENET news groups:
comp.protocols.dns.bind

For BIND mailing lists, see:

- http://www.isc.org/ml-archives/
  - BIND Users
    - Subscribe by sending mail to bind-users-request@isc.org
    - Submit questions or answers to this forum by sending mail to bind-users@isc.org
  - BIND 9 Users (Note: This list may not be maintained indefinitely.)
    - Subscribe by sending mail to bind9-users-request@isc.org
    - Submit questions or answers to this forum by sending mail to bind9-users@isc.org

For definitions of the terms and abbreviations used in this book, you can view or download the latest *IBM Glossary of Computing Terms* at the following Web address:

http://www.ibm.com/ibm/terminology

Note: Any pointers in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Licensed Documents

z/OS Communications Server licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at http://www.ibm.com/servers/resourcelink. Licensed books are available only to customers with a z/OS Communications Server license. Access to these books requires an IBM Resource Link Web user ID and password, and a key code. With your z/OS Communications Server order, you received a memo that includes this key code. To obtain your IBM Resource Link Web user ID and password, log on to http://www.ibm.com/servers/resourcelink. To register for access to the z/OS licensed books perform the following steps:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **User Profiles** located on the left-hand navigation bar.

3. Click on **Access Profile**.
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code, you will receive confirmation that your request is being processed. After your request is processed, you will receive an e-mail confirmation.

You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed. To access the licensed books:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **Library.**
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS Communications Server**.
6. Access the licensed book by selecting the appropriate element.

LookAt, an Online Message Help Facility

LookAt is an online facility that allows you to look up explanations for z/OS CS messages and system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on **News and Help** or from the z/OS V1R2 Collection, SK3T-4269.

To find a message explanation from a TSO command line, simply enter **lookat+message ID**, as in the following example:

```
lookat ezz8477i
```

This results in direct access to the message explanation for message EZZ8477I.

You can use LookAt on the Internet at the following Web site:
www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html

To find a message explanation from the LookAt Web site, simply enter the message ID. You can select the release, if applicable.

How to Contact IBM Service

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

z/OS Communications Server Information

This section contains descriptions of the books in the z/OS Communications Server library.

z/OS Communications Server publications are available:

- Online at the z/OS Internet Library web page at http://www.ibm.com/servers/eserver/zseries/zos/
- In hardcopy and softcopy
- In softcopy only

Softcopy Information

Softcopy publications are available in the following collections:

Titles	Order Number	Description
z/OS V1R2 Collection	SK3T-4269	This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R2, in both BookManager and PDF formats.
z/OS Software Products Collection	SK3T-4270	This CD includes, in both BookManager and PDF formats, the libraries of z/OS software products that run on z/OS but are not elements and features, as well as the *Getting Started with Parallel Sysplex* bookshelf.
z/OS V1R2 and Software Products DVD Collection	SK3T-4271	This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager and PDF format. This collection combines SK3T-4269 and SK3T-4270.
z/OS Licensed Product Library	SK3T-4307	This CD includes the licensed books in both BookManager and PDF format.
System Center Publication IBM S/390 Redbooks Collection	SK2T-2177	This collection contains over 300 ITSO redbooks that apply to the S/390 platform and to host networking arranged into subject bookshelves.

z/OS Communications Server Library

The following abbreviations follow each order number in the tables below.

HC/SC — Both hardcopy and softcopy are available.

SC — Only softcopy is available. These books are available on the CD Rom accompanying z/OS (SK3T-4269 or SK3T-4307). Unlicensed books can be viewed at the z/OS Internet library site.

Updates to books are available on RETAIN and in the document called *OS/390 DOC APARs and ++HOLD DOC data* which can be found at http://www.s390.ibm.com/os390/bkserv/ new_tech_info.html. See “Appendix A. Information Apars” on page 677 for a list of the books and the informational apars (info apars) associated with them.

Planning and Migration:

Title	Number	Format	Description
z/OS Communications Server: SNA Migration	GC31-8774	HC/SC	This book is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.

Title	Number	Format	Description
z/OS Communications Server: IP Migration	GC31-8773	HC/SC	This book is intended to help you plan for TCP/IP Services, whether you are migrating from a previous version or installing IP for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with TCP/IP Services.

Resource Definition, Configuration, and Tuning:

Title	Number	Format	Description
z/OS Communications Server: IP Configuration Guide	SC31-8775	HC/SC	This book describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this book in conjunction with the *z/OS Communications Server: IP Configuration Reference*.
z/OS Communications Server: IP Configuration Reference	SC31-8776	HC/SC	This book presents information for people who want to administer and maintain IP. Use this book in conjunction with the *z/OS Communications Server: IP Configuration Guide*. The information in this book includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • SMF records • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC31-8777	HC/SC	This book presents the major concepts involved in implementing an SNA network. Use this book in conjunction with the *z/OS Communications Server: SNA Resource Definition Reference*.
z/OS Communications Server: SNA Resource Definition Reference	SC31-8778	HC/SC	This book describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA.Use this book in conjunction with the *z/OS Communications Server: SNA Network Implementation Guide*.
z/OS Communications Server: SNA Resource Definition Samples	SC31-8836	SC	This book contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: AnyNet SNA over TCP/IP	SC31-8832	SC	This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.
z/OS Communications Server: AnyNet Sockets over SNA	SC31-8831	SC	This guide provides information to help you install, configure, use, and diagnose sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.

Operation:

Title	Number	Format	Description
z/OS Communications Server: IP User's Guide and Commands	SC31-8780	HC/SC	This book describes how to use TCP/IP applications. It contains requests that allow a user to: log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC31-8781	HC/SC	This book describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
z/OS Communications Server: SNA Operation	SC31-8779	HC/SC	This book serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SX75-0124	HC/SC	This book contains essential information about SNA and IP commands.

Customization:

Title	Number	Format	Description
z/OS Communications Server: SNA Customization	LY43-0092	SC	This book enables you to customize SNA, and includes the following: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines
z/OS Communications Server: IP Network Print Facility	SC31-8833	SC	This book is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Writing Application Programs:

Title	Number	Format	Description
z/OS Communications Server: IP Application Programming Interface Guide	SC31-8788	SC	This book describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC31-8807	SC	This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC31-8830	SC	This book is for programmers who want application programs that use the IMS TCP/IP application development services provided by IBM's TCP/IP Services.

Title	Number	Format	Description
z/OS Communications Server: IP Programmer's Reference	SC31-8787	SC	This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC31-8829	SC	This book describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC31-8811	SC	This book describes how to use the SNA LU 6.2 application programming interface for host application programs. This book applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this book.)
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC31-8810	SC	This book provides reference material for the SNA LU 6.2 programming interface for host application programs.
z/OS Communications Server: CSM Guide	SC31-8808	SC	This book describes how applications use the communications storage manager.
z/OS Communications Server: CMIP Services and Topology Agent Guide	SC31-8828	SC	This book describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The book provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis:

Title	Number	Format	Description
z/OS Communications Server: IP Diagnosis	GC31-8782	HC/SC	This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: SNA Diagnosis Vol 1 Techniques and Procedures and *z/OS Communications Server: SNA Diagnosis Vol 2 FFST Dumps and the VIT*	LY43-0088 LY43-0089	HC/SC	These books help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
z/OS Communications Server: SNA Data Areas Volume 1 and *z/OS Communications Server: SNA Data Areas Volume 2*	LY43-0090 LY43-0091	SC	These books describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and Codes:

Title	Number	Format	Description
z/OS Communications Server: SNA Messages	SC31-8790	HC/SC	This book describes the ELM, IKT, IST, ISU, IUT, IVT, and USS messages. Other information in this book includes: • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
z/OS Communications Server: IP Messages Volume 1 (EZA)	SC31-8783	HC/SC	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB)	SC31-8784	HC/SC	This volume contains TCP/IP messages beginning with EZB.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC31-8785	HC/SC	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ-SNM)	SC31-8786	HC/SC	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC31-8791	HC/SC	This book describes codes and other information that appear in z/OS Communications Server messages.

APPC Application Suite:

Title	Number	Format	Description
z/OS Communications Server: APPC Application Suite User's Guide	GC31-8809	SC	This book documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC application suite. Although its primary audience is the end user, administrators and application programmers may also find it useful.
z/OS Communications Server: APPC Application Suite Administration	SC31-8835	SC	This book contains the information that administrators need to configure the APPC application suite and to manage the APING, ANAME, AFTP, and A3270 servers.
z/OS Communications Server: APPC Application Suite Programming	SC31-8834	SC	This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Redbooks

The following Redbooks may help you as you implement z/OS Communications Server.

Title	Number
TCP/IP Tutorial and Technical Overview	GG24–3376
SNA and TCP/IP Integration	SG24–5291
IBM Communication Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 1: Configuration and Routing	SG24–5227
IBM Communication Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 2: UNIX Applications	SG24–5228
IBM Communication Server for OS/390 V2R10 TCP/IP Implementation Guide: Volume 3: MVS Applications	SG24–5229
OS/390 Secureway Communication Server V2R8 TCP/IP Guide to Enhancements	SG24–5631

Title	Number
TCP/IP in a Sysplex	SG24–5235
Managing OS/390 TCP/IP with SNMP	SG24–5866
Security in OS/390–based TCP/IP Networks	SG24–5383
IP Network Design Guide	SG24–2580

Related Information

For information about z/OS products, refer to *z/OS Information Roadmap* (SA22-7500). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, as well as describing each z/OS publication.

The table below lists books that may be helpful to readers.

Title	Number
z/OS SecureWay Security Server Firewall Technologies	SC24-5922
S/390: OSA-Express Customer's Guide and Reference	SA22-7403
z/OS MVS Diagnosis: Procedures	GA22-7587
z/OS MVS Diagnosis: Reference	GA22-7588
z/OS MVS Diagnosis: Tools and Service Aids	GA22-7589

Determining If a Publication Is Current

As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager softcopy are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. Here is how to determine if you are looking at the most current copy of a publication:

1. At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.
2. If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.
3. To compare softcopy publications, you can check the last two characters of the publication's filename (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

Summary of Changes

Summary of Changes
for SC31–8810–00
z/OS Version 1 Release 2

This book contains information previously presented in *OS/390 V2R5 eNetwork Communications Server: SNA Programmer's LU 6.2 Reference*, SC31-8568.

New Information

- Added the input parameter LUAFFIN to the following macroinstructions:
  - "APPCCMD CONTROL=ALLOC, QUALIFY=ALLOCD" on page 7.
  - "APPCCMD CONTROL=ALLOC, QUALIFY=CONVGRP" on page 19.
  - "APPCCMD CONTROL=ALLOC, QUALIFY=CONWIN" on page 31.
  - "APPCCMD CONTROL=ALLOC, QUALIFY=WHENFREE" on page 54.
  - "APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS" on page 150.
  - "APPCCMD CONTROL=PREALLOC, QUALIFY=ALLOCD" on page 186.
  - "APPCCMD CONTROL=PREALLOC, QUALIFY=CONVGRP" on page 197.
  - "APPCCMD CONTROL=PREALLOC, QUALIFY=CONWIN" on page 208.
  - "APPCCMD CONTROL=PREALLOC, QUALIFY=WHENFREE" on page 230.
  - "ISTRPL6" on page 575.
- Added RPL6AFFN to the 00002F offset within the ISTRPL6X DSECT. See "RPL Extension (ISTRPL6X)" on page 637.
- Added flag byte — RPL6FLG4 (LUAFFIN) to ″Layout of the RPL Extension″. See Figure 2 on page 643.

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

© Copyright IBM Corp. 1988, 2001

Chapter 1. LU 6.2 Macroinstruction Syntax and Operands

About This Chapter

This chapter describes all varieties of the VTAM APPCCMD macroinstruction. Separate descriptions are included for each CONTROL and QUALIFY combination of the macroinstruction. This chapter also includes the ISTGAPPC and ISTRPL6 macroinstructions. Macroinstruction descriptions are arranged alphabetically.

The macroinstructions are coded as assembler instructions, using name, operation, and operand fields. Refer to the *IBM High Level Assembler Language Reference for MVS and VM* for coding guidelines.

How Macroinstructions Are Described

Each macroinstruction description contains:

- The purpose of the macroinstruction
- General comments about its use
- Reference to tutorial chapters detailing its use
- The syntax of the macroinstruction and all parameters
- Input parameter descriptions
- RPL and RPL extension fields set by the macroinstruction
- Return and reason codes that can be returned for the macroinstruction

Syntax Descriptions

The syntax for the macroinstruction is described using the following format:

```
►►──┬──────────┬──APPCCMD ┤ Required and Optional Operands ├──────────────────────►◄
    └─name─────┘
```

Note: If you are not familiar with this type of syntax diagram, see “How to Read the Syntax Diagrams” on page 4.

Name Field

The *name* field provides a label for the macroinstruction. If you use a name, it must begin in column one of the macroinstruction, be followed by one or more blanks, and contain one to eight characters in the following format:

First character	Alphabetical (A-Z) or the national characters @, #, or $
Second to eighth character	Alphabetical (A-Z), numerical (0-9), or the national characters @, #, or $

The Macroinstruction Field

The *APPCCMD* identifier is always coded exactly as shown. It begins in column ten and must be preceded and followed by one or more blanks.

Operand Field

Required and optional operands direct information to the expansion program in the assembler. Generally, the information provided by the operand is made part of a

© Copyright IBM Corp. 1988, 2001

parameter list provided to VTAM during program processing. All operands for a macroinstruction appear in the syntax diagram.

Operands can occupy columns 16-71. You must place one or more blanks after the last operand on a line. If the operands require more than one line, you must place a nonblank *continuation character* in column 72. (See “Coding Continued Lines” on page 4.)

Operands consist of a fixed character string (the operand keyword) followed by an equal sign (=) and one or more operand values. If the value is specified as *name*, it must follow the rules for a symbolic name detailed in *IBM High Level Assembler Language Reference for MVS and VM*.

Operands do not have to be coded in the order shown by the syntax diagram. For example, a macroinstruction having the operands `AREALEN=data_length` and `AREA=data_area_address` could be coded in either of two ways:

```
AREALEN=132,AREA=WORK

AREA=WORK,AREALEN=132
```

Keyword operands must be separated by commas. If you choose to omit a keyword operand, also omit the comma that would have been included with it.

When more than one value can be coded after the keyword, a parenthesis is required. The OPTCD keyword can specify the manner of processing (asynchronous or synchronous) and control use of the buffer list option. For example, code the following to specify synchronous and buffer list options:
`OPTCD=(SYN,BUFFLST)`

Operand Descriptions

Each operand description begins with an explanation of the operand function. If the operand allows more than one fixed value that can be supplied with it, the operand description explains the effect that each value has on the action performed by the macroinstruction.

Most operands are coded by these general rules. If the format varies from these rules, the format is included with the description.

- When a quantity is indicated (for example, RECLEN=*data_length*), you can specify the value with:
  - Decimal integers.
  - An expression that is equal to such a value (for example, RECLEN=TEN, where TEN EQU 5*2).
  - The number of the register (enclosed by parentheses) to contain the quantity when the macroinstruction is executed. When specifying a quantity, Register notation is restricted to registers 2-12.
- When an address is indicated (for example, AREA=*data_area_address*), you can use any expression valid for an RX-type assembler instruction (for example, an LA instruction). Registers 1-12 can be specified for any operand that designates the address of an RPL. Register notation for all other address operands is restricted to registers 2-12.

For more information on operand formats, refer to the assembler language publication appropriate to your operating system.

Completion Information

All executable macroinstructions pass return codes in registers, and most indicate status information in control block fields when they are posted complete. This status information is described at the end of the macroinstruction description. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for information regarding the sequence of error checking.

Coding Default Values

The default values apply only to declarative macroinstructions ACB, EXLST, NIB, ISTRPL6, and RPL. All other macroinstructions (including APPCCMDs) use values specified in the macroinstruction itself or currently defined in the referenced control blocks. APPCCMDs do not have defaults; the defaults are in the underlying RPL and RPL extension.

Coding Comments

Comments can contain any characters valid in the assembler language. You can write comments following the operand field, but they must meet the following criteria:

- Comments must begin in column 17 or beyond.
- Comments must be separated from the last operand in the field by one or more blanks.
- Comments must not extend into the continuation column, column 72.

Comments can be continued on more than one line by placing an asterisk (*) in column one as shown in Figure 1.

```
LABEL1     OP1          OPERAND1,OPERAND1,OPERAND3,OPERX
                        AND4,OPERAND5    THIS IS ONE WAY
*
LABEL2     OP2          OPERAND1,OPERAND2,  AND THIS          X
                        OPERAND3,OPERAND4   IS ANOTHER
*                                             WAY

|          |           |                                    |
|          |           |                                    |
|          |           |                                    |
|          |           |                                    |
| column1  | column 10 | column 16                          | column 72 |
```

Figure 1. How to Code Comments and Continuation Lines

Note: A macroinstruction that has no operands cannot have comments on APPCCMD identifier line.

An entire line can be used for a comment; place an asterisk in the first column of the line. If you need several entire lines for comments, place an asterisk in the first column of each line and leave column 72 blank.

In this manual, the comments field is not shown in the macroinstruction syntax diagram.

Coding Continued Lines

Code VTAM macroinstructions in columns 1-71 of a line. Macroinstructions that exceed 71 columns can be continued on additional lines as shown in Figure 1 on page 3. (Continuation characters are omitted from other examples in this manual.) When you need to continue on another line, the following steps apply:

1. Code the macroinstruction one of two ways:
   - Through column 71
   - Through any completed operand, stopping after the comma that separates the operand from those that follow
2. Enter a nonblank continuation character in column 72. The continuation character is not considered to be part of the macroinstruction.
3. Continue operands beginning in column 16 of the next line. Columns 1-15 must be blank. A continuation line that begins in column 17 or later is ignored. A comment line cannot follow a continuation line.
4. If you must continue on another line, proceed with Step 1.
5. Macroinstructions can be coded on as many lines as needed.
6. Comments can appear on every line of a continued macroinstruction.
7. Columns 73-80 can be used to code identification characters, macroinstruction sequence characters, or both.

How to Read the Syntax Diagrams

This section describes how to read the syntax diagrams used in this book.

- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads (►►) and ends on the right with two arrowheads facing each other (►◄).

```
►►─┤ Syntax Diagram ├──────────────────────────────────────────►◄
```

- If a diagram is longer than one line, the first line ends with a single arrowhead (►) and the second line begins with a single arrowhead.

```
►►─┤ First Line ├──────────────────────────────────────────────►◄

►►─┤ Second Line ├─────────────────────────────────────────────►◄
```

- Required operands and values appear on the main path line.

```
►►──REQUIRED_OPERAND──────────────────────────────────────────►◄
```

  You must code required operands and values.
  If your choices are greater than one, the choices are stacked vertically in alphanumeric order.

```
►►──┬─REQUIRED_OPERAND_OR_VALUE_1─┬──────────────────────────────────►◄
    └─REQUIRED_OPERAND_OR_VALUE_2─┘
```

- Optional operands and values appear below the main path line.

```
►►──┬──────────────┬──────────────────────────────────────────────────►◄
    └─OPERAND──────┘
```

You can choose not to code optional operands and values.
If your choices are more than one, they are stacked vertically in alphanumeric order below the main path line.

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    ├─OPERAND_OR_VALUE_1───┤
    └─OPERAND_OR_VALUE_2───┘
```

- An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The comma means that each operand or value must be separated from the next by a comma.

```
          ┌─,──────────────────┐
          ▼                    │
►►──────REPEATABLE_OPERAND─────┴──────────────────────────────────────►◄
```

- An arrow returning to the left above a group of operands or values means that more than one can be selected, or a single one can be repeated.

```
►►──┬──────────────────────────────────────────────────────────────────┬──►◄
    │      ┌─,──────────────────────────────────┐                      │
    │      ▼                                    │                      │
    └────────┬─REPEATABLE_OPERAND_OR_VALUE_1─┬──┴──────────────────────┘
             └─REPEATABLE_OPERAND_OR_VALUE_2─┘
```

- A word in all uppercase is an operand or value you must spell exactly as shown. In this example, you must code ***OPERAND***.

  Note: VTAM commands are not case-sensitive. You can code them in uppercase or lowercase.

```
►►──OPERAND──────────────────────────────────────────────────────────►◄
```

If an operand or value can be abbreviated, the abbreviation is discussed in the text associated with the syntax diagram.

- If a diagram shows a character that is not alphanumeric (parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code ***OPERAND=(001,0.001)***.

```
►►──OPERAND=(001,0.001)──────────────────────────────────────────────►◄
```

- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code ***OPERAND=(001 FIXED)***.

```
►►──OPERAND=(001 FIXED)──────────────────────────────────────────────────►◄
```

- Default operands and values appear above the main path line. VTAM uses the default if you omit the operand entirely.

```
      ┌─DEFAULT─┐
►►────┤         ├──────────────────────────────────────────────────────────►◄
      └─OPERAND─┘
```

- A word appearing in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

```
►►──variable─────────────────────────────────────────────────────────────►◄
```

- References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

```
            (1)
►►──OPERAND──────────────────────────────────────────────────────────────►◄
```

Notes:

1 An example of a syntax note.

- Some diagrams contain syntax fragments; these serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names appear in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

```
►►──┤ Reference to Syntax Fragment ├──────────────────────────────────────►◄
```

Syntax Fragment:

```
├──1ST_OPERAND,2ND_OPERAND,3RD_OPERAND──────────────────────────────────┤
```

APPCCMD CONTROL=ALLOC, QUALIFY=ALLOCD

Purpose

This macroinstruction allocates resources for a conversation and assigns a contention winner or contention loser session to the conversation. If a session is not available and session limits allow, VTAM activates a session for the conversation, if possible.

Usage

QUALIFY=ALLOCD is used when an application program allocates a conversation and wants VTAM to queue the request if the request cannot be met immediately. This macroinstruction corresponds to the ALLOCATE RETURN_CONTROL (WHEN_SESSION_ALLOCATED) verb in the LU 6.2 architecture. This macroinstruction completes when VTAM assigns a session to the conversation or when an error occurs that prevents VTAM from assigning a session.

VTAM finds a session for the conversation as follows:

- If a session is free, VTAM assigns it to the conversation.
- If no free sessions exist and session limits allow, VTAM establishes a session and assigns it to the conversation.
- If a new session cannot be established, VTAM queues the request until a session becomes available or until the session limits are changed to allow the establishment of a new session.

When a conversation is allocated, VTAM assigns a conversation identifier to it. This identifier is returned in the CONVID field. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received on this conversation.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for details on allocating a conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is suspended for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
└──────┘──APPCCMD CONTROL=ALLOC──,QUALIFY=ALLOCD──────(2)──,RPL=──┬─rpl_address_field──────────(8)──┬──►
└name ─┘                                                          └─(rpl_address_register)──────┘
```

```
►──┬──────────────
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                 (2)
   └─,RTSRTRN=──┬─BOTH─┬──────────────┬──────────────────────────────────────────────────────────────────────────────────
                └─EXPD─┘              │                                                                                   │
                                      └─,USERFLD=──┬─4-bytes_of_user_data─┬──────────────────────────────────────────────┘
                                                   └─(user_data_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                   (2)
   └─,VTRINA=──┬─vector_address_field─────────────┬──────────────────────────────────────────────────────────────────┘
               └─(vector_address_register)────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                                                                                   (2)
   └─,VTRINL=──┬─vector_length_field─────────────┬──────────────────────────────────────────────────────────────────┘
               └─(vector_length_register)────────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*fmh-5_and_opt._pip_gds_var._add._field*

AREA=(*fmh-5_and_opt._pip_gds_var._add._reg.***)**
Specifies the address of the data area containing a formatted FMH-5. A formatted GDS field can follow the FMH-5 in the data area. See “FMH-5 (ISTFM5)” on page 635 for the VTAM-supplied DSECT that can be used to create and test values for an FMH-5. Refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for a description of the FMH-5 and GDS variable. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NAMEUSE
Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME
The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME
The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*fmh-5_and_opt._pip_gds_var._len.*

RECLEN=(*fmh-5_and_opt._pip_gds_var._len._reg.***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in

the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4-bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following values for half-duplex conversations:

X'00' RESET
X'01' SEND
X'08' END_CONVERSATION

This field can have the following values for full-duplex conversations:
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5

has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed

because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversations state is SEND after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_NAME_REQUESTED

RCPRI	RCSEC	Meaning
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'0006'	ALLOCATION_ERROR—SYNCLEVEL_NOT_SUPPORTED_ BY_LU
X'0004'	X'0010'	ALLOCATION_ERROR—SYNCLEVEL_NOT_VALID_ FOR_FULL_DUPLEX
X'0004'	X'0011'	ALLOCATION_ERROR—LU_PAIR_NOT_SUPPORTING_ FULL_DUPLEX_CONVERSATIONS
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000A'	PARAMETER_ERROR—INCOMPLETE_FMH5_SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0015'	PARAMETER_ERROR—INVALID_TPN
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE.
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=ALLOC, QUALIFY=CONVGRP

Purpose

This macroinstruction allocates resources for a conversation and assigns to the conversation a session with a specified conversation group identifier. If the specific session is not available and session limits allow, VTAM queues the request until the session becomes available. If the specific session does not exist, VTAM fails the allocation request.

Usage

QUALIFY=CONVGRP is used to allocate a conversation over a specific session that already exists. It provides the ability to serially allocate a related group of conversations on a particular session. This macroinstruction corresponds to the ALLOCATE RETURN_ CONTROL (WHEN_CONVERSATION_GROUP_ALLOCATED) verb in the LU 6.2 architecture. This macroinstruction completes when:

- VTAM assigns the specified session to the conversation.
- The specific session is deactivated.
- An error occurs that prevents VTAM from assigning the session to the conversation.

To indicate the session to be used, the application program specifies the conversation group identifier for the session on the CGID keyword. This value was returned to the application program by the CGID returned field for a previous APPCCMD CONTROL=ALLOC, CONTROL=PREALLOC, or CONTROL=RCVFMH5 macroinstruction.

VTAM finds the session for the conversation as follows:

- If the specified session is available, VTAM assigns it to the conversation.
- If the specified session exists but is not available, VTAM queues the request until the session becomes available.
- If the specified session is deactivated while the request is queued, the queued request will be rejected.

As with other ALLOC macroinstructions, when the conversation is allocated, VTAM assigns a conversation identifier to it. This identifier is returned in the CONVID field. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for details on allocating a conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
  [name] APPCCMD CONTROL=ALLOC ,QUALIFY=CONVGRP (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_field (1)
        (acb_address_register)
►
  ,AREA= fmh-5_and_opt._pip_gds_var._add._field (1)
         (fmh-5_and_opt._pip_gds_var._add._reg.)
  ,BRANCH= NO (1)
           YES
►
  ,CGID= 32-bit_conversation_group_id_field (2)
         (32-bit_conversation_group_id_register)
►
  ,CONMODE= BUFFCA (2)
            CS
            LLCA
  ,CONXMOD= CA (2)
            CS
►
  ,ECB= INTERNAL (3) (1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field (4) (1)
         (exit_routine_address_register)
►
  ,LUAFFIN= APPL (2)
            NOTAPPL
  ,NAMEUSE= APNAME (2)
            GNAME
►
  ,OPTCD=( ,
           (7)
           ASY (1)
           SYN (1)
           KEEPSRB (1) (5)
           NKEEPSRB (1) (6) )
►
  ,RECLEN= fmh-5_and_opt._pip_gds_var._len. (1)
           (fmh-5_and_opt._pip_gds_var._len._reg.)
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                 (2)
   └─,RTSRTRN=──┬─BOTH─┬──────────────┘   └─,USERFLD=──┬─4-bytes_of_user_data─┬──────────────┘
                └─EXPD─┘                                └─(user_data_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                    (2)
   └─,VTRINA=──┬─vector_address_field─────────┬──────────────┘
               └─(vector_address_register)────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                                                                                    (2)
   └─,VTRINL=──┬─vector_length_field─────────┬──────────────┘
               └─(vector_length_register)────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*fmh-5_and_opt._pip_gds_var._add._field*

AREA=(*fmh-5_and_opt._pip_gds_var._add._reg.***)**
Specifies the address of the data area containing a formatted FMH-5. A formatted GDS field can follow the FMH-5 in the data area. See “FMH-5

(ISTFM5)" on page 635 for the VTAM-supplied DSECT that can be used to create and test values for an FMH-5. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the FMH-5 and GDS variable. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CGID=*32-bit_conversation_group_id_field*

CGID=(*32-bit_conversation_group_id_register***)**
Specifies the 32-bit conversation group ID.

This value can be obtained from a previous APPCCMD CONTROL=ALLOC, CONTROL=PREALLOC, or CONTROL=RCVFMH5 macroinstruction. If the CGID operand is not specified, VTAM uses the conversation group ID that is already in the RPL6CGID field on the RPL extension.

The conversation group ID identifies a specific session between two specific LUs. It provides a means by which a VTAM LU 6.2 application program and its partner LU can share serially the same session.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE,

QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

NAMEUSE
Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME
The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME
The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*fmh-5_and_opt._pip_gds_var._len.*

RECLEN=(*fmh-5_and_opt._pip_gds_var._len._reg.***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields.

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CONSTATE
The field in the RPL extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

This field can have the following values for half-duplex conversations:
X'00' RESET
X'01' SEND
X'08' END_CONVERSATION

This field can have the following values for full-duplex conversations:
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGMODE
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

LUNAME
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NETID
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located.

This network identifier is the identifier of the partner LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the

session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversation, the conversation state is SEND after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_ NAME_REQUESTED
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'0006'	ALLOCATION_ERROR—SYNCLEVEL_NOT_SUPPORTED_ BY_LU

RCPRI	RCSEC	Meaning
X'0004'	X'0010'	ALLOCATION_ERROR—SYNCLEVEL_NOT_VALID_ FOR_FULL_DUPLEX
X'0004'	X'0011'	ALLOCATION_ERROR—LU_PAIR_NOT_SUPPORTING_ FULL_DUPLEX_CONVERSATIONS
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'000A'	PARAMETER_ERROR—INCOMPLETE_FMH5_SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0015'	PARAMETER_ERROR—INVALID_TPN
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002A'	PARAMETER_ERROR—INVALID_CGID_VALUE_SPECIFIED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=ALLOC, QUALIFY=CONWIN

Purpose

This macroinstruction allocates resources for a conversation and, if session limits allow, assigns a contention-winner session to the conversation. If a contention-winner session is not available, VTAM queues the request for later completion.

Usage

QUALIFY=CONWIN is used when an application program allocates a conversation and wants VTAM to queue the request if no contention-winner session can be assigned. This macroinstruction corresponds to the ALLOCATE_RETURN_CONTROL (WHEN_CONTENTION WINNER_ALLOCATED) verb in the LU 6.2 architecture. This macroinstruction completes when VTAM assigns a contention-winner session or an error occurs that prevents VTAM from assigning a session.

VTAM finds a session for the conversation as follows:

- If a contention-winner session is currently available, VTAM assigns it to the conversation.
- If no contention-winner session is available and session limits allow, VTAM establishes a new contention-winner session and assigns it to the conversation.
- If a new contention-winner session cannot be established, VTAM queues the request until a contention-winner session is available or session limits are changed to allow a new contention-winner session to be activated.

For this macroinstruction to complete successfully, the session limits must define at least one contention-winner session.

If contention-winner sessions are deactivated under normal conditions and an APPCCMD CONTROL=ALLOC, QUALIFY=CONWIN request is queued, VTAM activates another contention-winner session to meet the queued request.

The application program can specify how expedited data is to be received.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for details on allocating a conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
┌──────┐
└name  ┘──APPCCMD CONTROL=ALLOC──,QUALIFY=CONWIN──────(2)──,RPL=──┬─rpl_address_field──────────(8)──┬──►
                                                                  └─(rpl_address_register)─────────┘
```

```
►──┬──────────────
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                
```

AREA=(*fmh-5_and_opt._pip_gds_var._add._reg.***)**
Specifies the address of the data area containing a formatted FMH-5. A formatted GDS field can follow the FMH-5 in the data area. See “FMH-5 (ISTFM5)” on page 635 for the VTAM-supplied DSECT that can be used to create and test values for an FMH-5. Refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for a description of the FMH-5 and GDS variable. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NAMEUSE
Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME
The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME
The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*fmh-5_and_opt._pip_gds_var._len.*

RECLEN=(*fmh-5_and_opt._pip_gds_var._len._reg.***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in

the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

This field can have the following values for half-duplex conversations:

X'00' RESET
X'01' SEND
X'08' END_CONVERSATION

In addition to the half-duplex conversation states, this field can contain the following full-duplex conversation states:
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5

has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed

because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is SEND state after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See the "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_NAME_REQUESTED

RCPRI	RCSEC	Meaning
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'0006'	ALLOCATION_ERROR—SYNCLEVEL_NOT_SUPPORTED_ BY_LU
X'0004'	X'0010'	ALLOCATION_ERROR—SYNCLEVEL_NOT_VALID_ FOR_FULL_DUPLEX
X'0004'	X'0011'	ALLOCATION_ERROR—LU_PAIR_NOT_SUPPORTING_ FULL_DUPLEX_CONVERSATIONS
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000A'	PARAMETER_ERROR—INCOMPLETE_FMH5_SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0015'	PARAMETER_ERROR—INVALID_TPN
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=ALLOC, QUALIFY=IMMED

Purpose

This macroinstruction allocates resources for a conversation and if session limits allow, assigns an active contention-winner session to it. If no session is available, the allocation request fails.

Usage

QUALIFY=IMMED is used to allocate a conversation when the application program needs an immediate response from VTAM. This macroinstruction completes successfully only when an active contention-winner session is available to be assigned to the conversation. If the request cannot be met immediately, VTAM does not queue it. VTAM neither tries to activate a new session nor bids on a contention-loser session. APPCCMD CONTROL=ALLOC, QUALIFY=IMMED corresponds to the ALLOCATE RETURN_CONTROL(IMMEDIATE) verb in the LU 6.2 architecture.

When a conversation is allocated, VTAM assigns a conversation identifier to it. This identifier is returned in the CONVID field. The application program must associate a conversation with a particular transaction by using the conversation identifier.

The application program can specify how expedited data is to be received.

For details on allocating a conversation, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
 [name] APPCCMD CONTROL=ALLOC ,QUALIFY=IMMED(2) ,RPL= rpl_address_field(8)
                                                      (rpl_address_register)
►──────────────────────────────────────────────────────────────────────────►
 ,AAREA= rpl_extension_address_field(1)
         (rpl_extension_address_register)
►──────────────────────────────────────────────────────────────────────────►
 ,ACB= acb_address_field(1)
       (acb_address_register)
►──────────────────────────────────────────────────────────────────────────►
 ,AREA= fmh-5_and_opt._pip_gds_var._add._field(1)   ,BRANCH= NO(1)
        (fmh-5_and_opt._pip_gds_var._add._reg.)             YES
```

```
►──┬──────────────
```

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*fmh-5_and_opt._pip_gds_var._add._field*

AREA=(*fmh-5_and_opt._pip_gds_var._add._reg.***)**
Specifies the address of the data area containing a formatted FMH-5. A formatted GDS field can follow the FMH-5 in the data area. See "FMH-5 (ISTFM5)" on page 635 for the VTAM-supplied DSECT that can be used to create and test values for an FMH-5. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the FMH-5 and GDS variable. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running

under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE

Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA

Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS

Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA

Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD

Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB

Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL

Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*fmh-5_and_opt._pip_gds_var._len.*

RECLEN=(*fmh-5_and_opt._pip_gds_var._len._reg.***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD

CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

This field can have the following values for half-duplex conversations:
X'00' RESET
X'01' SEND
X'08' END_CONVERSATION

This field can have the following values for full-duplex conversations:
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is SEND after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See the “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0006'	ALLOCATION_ERROR—SYNCLEVEL_NOT_SUPPORTED_ BY_LU
X'0004'	X'0010'	ALLOCATION_ERROR—SYNCLEVEL_NOT_VALID_ FOR_FULL_DUPLEX
X'0004'	X'0011'	ALLOCATION_ERROR—LU_PAIR_NOT_SUPPORTING_ FULL_DUPLEX_CONVERSATIONS
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000A'	PARAMETER_ERROR—INCOMPLETE_FMH5_SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0015'	PARAMETER_ERROR—INVALID_TPN
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0058'	X'0000'	UNSUCCESSFUL,—SESSION_NOT_AVAILABLE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY

RCPRI	RCSEC	Meaning
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=ALLOC, QUALIFY=WHENFREE

Purpose

This macroinstruction allocates resources for a conversation and if session limits allow, assigns a session to the conversation. If a session is not available and one cannot be activated, VTAM returns control to the application program.

Usage

QUALIFY=WHENFREE is used when an application program allocates a conversation and wants VTAM to search for a session that satisfies the ALLOCATE request. This macroinstruction corresponds to the ALLOCATE RETURN_CONTROL (WHEN_SESSION_FREE) verb in the LU 6.2 architecture. This macroinstruction completes when VTAM assigns a session to the conversation or when VTAM cannot assign a session and returns control to the application program with a return code of X'0004', X'0001'.

VTAM finds a session for the conversation as follows:

1. If a session is available, VTAM assigns it to the conversation.
2. If no available sessions exist and session limits allow, VTAM establishes a session and assigns it to the conversation.
3. If a session cannot be established and session activation requests are pending, VTAM queues the ALLOCATE request until the request is satisfied or until all pending session activation requests are used. If the pending session activation requests are used before the ALLOCATE request is satisfied, VTAM fails the ALLOCATE request with an RCPRI, RCSEC code of X'0004', X'0001'.
4. If a session cannot be established and no session activation request is pending that might satisfy the ALLOCATE request, VTAM fails the ALLOCATE request with an RCPRI, RCSEC code of X'0004', X'0001' and returns control to the application program.

When a conversation is allocated, VTAM assigns a conversation identifier to it. This identifier is returned in the CONVID field. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for details on allocating a conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
 ┌──────────┐
 └─name─────┘─APPCCMD CONTROL=ALLOC──,QUALIFY=WHENFREE──────(2)──,RPL=─┬─rpl_address_field──────────────(8)─┬──►
                                                                        └─(rpl_address_register)──────────┘
```

```
►──┬──────────────
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────►
   │                                                   (2)
   └─,VTRINA=──┬─vector_address_field──────────────────┬──
               └─(vector_address_register)─────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                   (2)
   └─,VTRINL=──┬─vector_length_field──────────────────┬──
               └─(vector_length_register)─────────────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*fmh-5_and_opt._pip_gds_var._add._field*

AREA=(*fmh-5_and_opt._pip_gds_var._add._reg.***)**
Specifies the address of the data area containing a formatted FMH-5. A formatted GDS field can follow the FMH-5 in the data area. See “FMH-5 (ISTFM5)” on page 635 for the VTAM-supplied DSECT that can be used to create and test values for an FMH-5. Refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for a description of the FMH-5 and GDS variable. This field is labeled RPLAREA in the RPL.

BRANCH

Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO

Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES

Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE

Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA

Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS

Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA

Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD

Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS

Specifies that the mode for expedited information is to be put in such a

state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NAMEUSE
Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME
The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME
The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*fmh-5_and_opt._pip_gds_var._len.*

RECLEN=(*fmh-5_and_opt._pip_gds_var._len._reg.***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were

supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates which state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following values for half-duplex conversations:
X'00' RESET
X'01' SEND
X'08' END_CONVERSATION

This field can have the following values for full-duplex conversations:
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is SEND after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See the "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_ NAME_REQUESTED
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'0006'	ALLOCATION_ERROR—SYNCLEVEL_NOT_SUPPORTED_ BY_LU
X'0004'	X'0010'	ALLOCATION_ERROR—SYNCLEVEL_NOT_VALID_ FOR_FULL_DUPLEX
X'0004'	X'0011'	ALLOCATION_ERROR—LU_PAIR_NOT_SUPPORTING_ FULL_DUPLEX_CONVERSATIONS

RCPRI	RCSEC	Meaning
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000A'	PARAMETER_ERROR—INCOMPLETE_FMH5_SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0015'	PARAMETER_ERROR—INVALID_TPN
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=CHECK

Purpose

This macroinstruction waits for completion of an asynchronous macroinstruction request and marks the RPL and RPL extension used in the request as inactive upon completion.

Usage

When asynchronous handling is specified for an RPL-based request, the application program receives control when the request has been accepted by VTAM and the requested operation has been scheduled. An APPCCMD CONTROL=CHECK macroinstruction must be issued for the RPL used for the request to determine when the macroinstruction completes and to get the return code information. APPCCMD CONTROL=CHECK cannot be issued for synchronous requests. In addition, APPCCMD CONTROL=CHECK cannot be issued for an RPL that specifies a non-APPCCMD request. This macroinstruction can be issued in cross-memory mode against an active asynchronous RPL request *only* when the RPL's ECB has been posted or the RPL exit has been scheduled.

When APPCCMD CONTROL=CHECK is executed for an RPL that specifies an ECB, the following actions occur:

- If the operation being checked has not been completed, VTAM suspends the execution of the application program task under which the APPCCMD CONTROL=CHECK is issued until the operation is completed.
- If the operation being checked has completed, VTAM returns control to the next sequential instruction after the APPCCMD CONTROL=CHECK macroinstruction.
- The ECB (internal or external) is cleared before VTAM returns control to the application program. (The ECB must be cleared before an RPL-based macroinstruction is issued.)

  Note: The ECB specified in an asynchronous APPCCMD macroinstruction must not be cleared between the time it is issued and the corresponding APPCCMD CONTROL=CHECK is issued. If the ECB is cleared during this interval, control cannot be returned to the application program after the APPCCMD CONTROL=CHECK is issued.
- The RPL being checked is marked available for reuse by another APPCCMD macroinstruction. (APPCCMD CONTROL=CHECK is the only way this can be done for asynchronous APPCCMD requests.)

When APPCCMD CONTROL=CHECK is executed in an RPL exit routine for the associated RPL, the following actions occur:

- VTAM marks the RPL being checked as available for reuse by another APPCCMD macroinstruction.
- If the operation being checked has completed, VTAM returns control to the next sequential instruction after the APPCCMD CONTROL=CHECK.

Context

Input states are not applicable to this macroinstruction.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
                                                        (1)
►►──┬──────────┬──APPCCMD CONTROL=CHECK──,RPL=──┬─rpl_address_field──────────┬──────►◄
    └─ name────┘                                └─(rpl_address_register)─────┘
```

Notes:

1 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

After the APPCCMD CONTROL=CHECK macroinstruction has completed, the completion information returned is for the macroinstruction being checked. Refer to the description of the particular APPCCMD being checked for a list of the parameters that are returned to the application program.

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDPROG

Purpose

This macroinstruction deallocates a conversation when an application program has detected a transaction program error.

Usage

QUALIFY=ABNDPROG is used when the application program detects an error in a transaction program and that error prevents further useful communication on the conversation. It corresponds to DEALLOCATE TYPE=ABEND_PROG in the LU 6.2 architecture. If the conversation is in a sending state, the SEND buffer is flushed before the conversation is deallocated.

This macroinstruction, along with the other QUALIFY=ABND* forms, can be used to cancel an outstanding APPCCMD macroinstruction, which allows the application program to recover from hung transactions. The following macroinstructions cannot be canceled:

- APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC that has not received any data from the partner LU
- APPCCMD CONTROL=RECEIVE, QUALIFY=ANY that has not been matched to a conversation
- APPCCMD CONTROL=RCVFMH5, QUALIFY=NULL|QUEUE
- APPCCMD CONTROL=RESETRCV
- APPCCMD CONTROL=OPRCNTL
- APPCCMD CONTROL=REJECT, QUALIFY=CONV
- APPCCMD CONTROL=TESTSTAT, QUALIFY=ALL|IALL
- One of the abnormal deallocation macroinstructions
- A macroinstruction that is waiting for a response to a confirmation request
- A macroinstruction that is waiting for the arrival of an FMH-7

If any one of these macroinstructions is outstanding, the application program must wait for it to complete before issuing this macroinstruction. Alternatively, the application program can issue an APPCCMD CONTROL=REJECT, QUALIFY=CONV macroinstruction.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for information on deallocating a conversation when an error is detected or for early deallocation of a pending APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5, QUALIFY=DATAQUE.

Context

On half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- PENDING_ALLOCATE
- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PEND_SEND
- PEND_END_CONV_LOG
- PENDING_RECEIVE_LOG

On full-duplex conversations, this macroinstruction can be issued from the following conversation states:

- PENDING_ALLOCATE
- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=ABNDPROG (2) ,RPL= rpl_address_field (8)
                                                            (rpl_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,AAREA= rpl_extension_address_field (1)
   │         (rpl_extension_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,ACB= acb_address_field (1)
   │       (acb_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,AREA= optional_log_data_area_address_field (1)   ,BRANCH= NO (1)
   │        (optional_log_data_area_address_register)          YES
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,CONVID= 32-bit_resource_id_field (2)
   │          (32-bit_resource_id_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,ECB= INTERNAL (3) (1)
   │       ecb_address_field
   │       (ecb_address_register)
   │ ,EXIT= exit_routine_address_field (4) (1)
   │        (exit_routine_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,OPTCD=( ,──────────────────────────────────────────────────────────────────────
   │          │ (7)                                                                  │
   │          │ ASY (1)                                                              │
   │          │ SYN (1)                                                              │
   │          │ KEEPSRB (1) (5)                                                      │
   │          │ NKEEPSRB (1) (6)                                                     │
   │          )
```

```
►──┬──────────────
```

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the

completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*optional_log_data_length*

RECLEN=(*optional_log_data_length_register***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL. A value of 0 in the RECLEN field indicates that the application program has chosen not to provide optional error log data to VTAM. If the application program specifies RECLEN=0, VTAM indicates in the FMH-7 it creates as a result of this APPCCMD that no error log data follows the FMH-7, and the AREA field in the RPL is ignored.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields.

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be

received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and

displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is END_CONV after successful completion of the macroinstruction. For full-duplex conversations, the conversation state is FDX_RESET after successful completion of the macroinstruction.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000B'	PARAMETER_ERROR—INCOMPLETE_GDS_VARIABLE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0021'	PARAMETER_ERROR—ABNORMAL_DEALLOCATE_ REJECTED_RETRY
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR

RCPRI	RCSEC	Meaning
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDSERV

Purpose

This macroinstruction is used when the application program detects an error in its implementation of LU 6.2 services.

Usage

QUALIFY=ABNDSERV is used when the application program encounters errors related to LU 6.2 services. For example, the application program might detect an error in its support of mapped conversations or in conversation-level security that would require it to deallocate the conversation. QUALIFY=ABNDSERV corresponds to the DEALLOCATE TYPE=ABEND_SVC verb in the LU 6.2 architecture.

If the conversation is in a state that allows sending, the function of the APPCCMD CONTROL=SEND, QUALIFY=FLUSH macroinstruction is executed prior to deallocating the conversation.

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDSERV can be issued against a conversation for which there is already an APPCCMD outstanding. These commands cancel the previous macroinstruction, allowing the application program to recover from a “hung” transaction. However, there are cases where it is not allowed when a prior macroinstruction is outstanding. See “Usage” on page 68 for a list of macroinstructions that cannot be canceled.

Refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for information on abnormally deallocating a conversation.

Context

On half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- PENDING_ALLOCATE
- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PEND_SEND
- PEND_END_CONV_LOG
- PENDING_RECEIVE_LOG

On full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- PENDING_ALLOCATE
- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
  [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=ABNDSERV (2) ,RPL= rpl_address_field (8)
                                                            (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_field (1)
        (acb_address_register)
►
  ,AREA= optional_log_data_area_address_field (1)
         (optional_log_data_area_address_register)
  ,BRANCH= NO (1)
           YES
►
  ,CONVID= 32-bit_resource_id_field (2)
           (32-bit_resource_id_register)
►
  ,ECB= INTERNAL (3) (1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field (4) (1)
         (exit_routine_address_register)
►
  ,OPTCD=( ,
           (7)
           ASY (1)
               (1)
           SYN
           KEEPSRB (1) (5)
               (1) (6)
           NKEEPSRB )
►
  ,RECLEN= optional_log_data_length (1)
           (optional_log_data_length_register)
►◄
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*optional_log_data_area_address_field*

AREA=*optional_log_data_area_address_field*
Specifies the address of a data area containing a formatted error log GDS variable to be sent to the partner application program. The application program is responsible for placing the error log data into the local system log. If the application program chooses to supply an error log GDS variable, it has to supply the entire GDS variable on the APPCCMD macroinstruction. VTAM inspects the 2-byte logical-record length (LL) field of the GDS variable to determine if the amount of data supplied is equal to the length specified in the LL field. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the error log GDS variable.) This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*optional_log_data_length*

RECLEN=(*optional_log_data_length_register***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL. A value of 0 in the RECLEN field indicates that

the application program has chosen not to provide optional error log data to VTAM. If the application program specifies RECLEN=0, VTAM indicates in the FMH-7 it creates as a result of this APPCCMD that no error log data follows the FMH-7, and the AREA field in the RPL is ignored.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the PRL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5

has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

The changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is END_CONV after successful processing.

For full-duplex conversations, the conversation state is FDX_RESET after successful processing.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000B'	PARAMETER_ERROR—INCOMPLETE_GDS_VARIABLE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0021'	PARAMETER_ERROR—ABNORMAL_DEALLOCATE_ REJECTED_RETRY
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDTIME

Purpose

This macroinstruction is used to deallocate a conversation that has had no activity for a specified amount of time.

Usage

QUALIFY=ABNDTIME is used when the LU detects that it has not received information from one of its transaction programs within a specific amount of time. For example, an application program would use this macroinstruction if one of the conversations is in a state that allows receiving and has not received any data in an excessive amount of time. The application program must determine how long to wait before issuing this macroinstruction.

If the conversation is in a state that allows sending, the function of the APPCCMD CONTROL=SEND, QUALIFY=FLUSH macroinstruction is executed prior to abnormally deallocating the conversation.

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDTIME can be issued against a conversation for which there is already an APPCCMD outstanding. These commands cancel the previous macroinstruction, allowing the application program to recover from a hung transaction. However, there are cases where it is not allowed when a prior macroinstruction is outstanding. See “Usage” on page 68 for a list of macroinstructions that cannot be canceled.

QUALIFY=ABNDTIME corresponds to the DEALLOCATE TYPE=ABEND_TIMER verb in the LU 6.2 architecture.

Refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for a description of abnormally terminating a conversation.

Context

On half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- PENDING_ALLOCATE
- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PEND_SEND
- PEND_END_CONV_LOG
- PENDING_RECEIVE_LOG

On full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- PENDING_ALLOCATE
- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
  [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=ABNDTIME (2) ,RPL= rpl_address_field (8)
                                                            (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_field (1)
        (acb_address_register)
►
  ,AREA= optional_log_data_area_address_field (1)
         (optional_log_data_area_address_register)
  ,BRANCH= NO (1)
           YES
►
  ,CONVID= 32-bit_resource_id_field (2)
           (32-bit_resource_id_register)
►
  ,ECB= INTERNAL (3) (1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field (4) (1)
         (exit_routine_address_register)
►
  ,OPTCD=( ,
           (7)
           ASY (1)
           SYN (1)
           KEEPSRB (1) (5)
           NKEEPSRB (1) (6)
           )
►
  ,RECLEN= optional_log_data_length (1)
           (optional_log_data_length_register)
►◄
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*optional_log_data_area_address_field*

AREA=(*optional_log_data_area_address_register***)**
Specifies the address of a data area containing a formatted error log GDS variable to be sent to the partner application program. The application program is responsible for placing the error log data into the local system log. If the application program chooses to supply an error log GDS variable, it has to supply the entire GDS variable on the APPCCMD macroinstruction. VTAM inspects the 2-byte logical-record length (LL) field of the GDS variable to determine if the amount of data supplied is equal to the length specified in the LL field. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the error log GDS variable.) This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running

under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*optional_log_data_length*

RECLEN=(*optional_log_data_length_register***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL. A value of 0 in the RECLEN field indicates that the application program has chosen not to provide optional error log data to VTAM. If the application program specifies RECLEN=0, VTAM indicates in the FMH-7 it creates as a result of this APPCCMD that no error log data follows the FMH-7, and the AREA field in the RPL is ignored.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6AVFA in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by

the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD

CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is END_CONV after successful processing.

For full-duplex conversations, the conversation state is FDX_RESET after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000B'	PARAMETER_ERROR—INCOMPLETE_GDS_VARIABLE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ ADDRESS_OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0021'	PARAMETER_ERROR—ABNORMAL_DEALLOCATE_ REJECTED_RETRY
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDUSER

Purpose

This macroinstruction deallocates a conversation when the application program detects an error.

Usage

This macroinstruction is used by an application program to deallocate a conversation and to inform the partner LU of the reason for the deallocation. To indicate the reason for the deallocation, the application program specifies a sense code on the macroinstruction. This sense code is sent to the partner LU in an FMH-7 and must be appropriate to the error. Otherwise improper processing of the macroinstruction might occur. For a list of valid sense codes, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

This macroinstruction does not correspond to any of the verbs in the LU 6.2 architecture.

An example of the use of this macroinstruction would be to report errors that the application program detects on a received FMH-5. Although VTAM performs preliminary format checks on the FMH-5 before passing it to the application program, the application program validates the FMH-5. If the application program detects an error in the FMH-5, it issues APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDUSER and specifies the appropriate sense code. VTAM sends the conversation deallocation notification and the FMH-7 to the partner LU.

If the conversation is in a state that allows sending, the function of the APPCCMD CONTROL=SEND, QUALIFY=FLUSH macroinstruction is executed prior to abnormally deallocating the conversation.

APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDUSER can be issued against a conversation for which there is already an APPCCMD outstanding. It cancels the previous macroinstruction, allowing the application program to recover from a "hung" transaction. However, in some cases, it is not allowed when a prior macroinstruction is outstanding. See "Usage" on page 68 for a list of macroinstructions that cannot be canceled.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information on abnormally deallocating a conversation.

Context

On half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- PENDING_ALLOCATE
- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PEND_SEND
- PEND_END_CONV_LOG
- PENDING_RECEIVE_LOG

On full-duplex conversations, this macroinstruction can be issued from the following conversation states:

- PENDING_ALLOCATE
- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=ABNDUSER (2) ,RPL= rpl_address_field (8)
                                                            (rpl_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,AARAEA= rpl_extension_address_field (1)
   │          (rpl_extension_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,ACB= acb_address_field (1)
   │       (acb_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,AREA= optional_log_data_area_address_field (1)   ┬──────────────────────────────►
   │        (optional_log_data_area_address_register)  │ ,BRANCH= NO (1)
   │                                                   │          YES
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,CONVID= 32-bit_resource_id_field (2)
   │          (32-bit_resource_id_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,ECB= INTERNAL (3) (1)
   │       ecb_address_field
   │       (ecb_address_register)
   │ ,EXIT= exit_routine_address_field (4) (1)
   │        (exit_routine_address_register)
►──┬──────────────────────────────────────────────────────────────────────────────────►
   │ ,OPTCD=( ,──────────────────────────────────────────────────────────────────────
   │          │ (7)
   │          │ ASY (1)
   │          │ SYN (1)
   │          │ KEEPSRB (1) (5)
   │          │ NKEEPSRB (1) (6)
   │          )
```

```
►──┬──────────────────────────────────────────────────────────────────────────►
   │                                                   (1)
   └,RECLEN=──┬─optional_log_data_length──────────────┬──────────────────────────
              └─(optional_log_data_length_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────►◄
   │                                                   (2)
   └,SENSE=──┬─user-supplied_32-bit_fmh-7_sense_code──────────────┬──────────────
             └─(user-supplied_32-bit_fmh-7_sense_code_register)─┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*optional_log_data_area_address_field*

AREA=(*optional_log_data_area_address_register***)**
Specifies the address of a data area containing a formatted error log GDS variable to be sent to the partner application program. The application program is responsible for placing the error log data into the local system log. If the application program chooses to supply an error log GDS variable, it has to supply the entire GDS variable on the APPCCMD macroinstruction. VTAM inspects the 2-byte logical-record length (LL) field of the GDS variable to

determine if the amount of data supplied is equal to the length specified in the LL field. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the error log GDS variable.) This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application

program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*optional_log_data_length*

RECLEN=(*optional_log_data_length_register***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL. A value of 0 in the RECLEN field indicates that the application program has chosen not to provide optional error log data to VTAM. If the application program specifies RECLEN=0, VTAM indicates in the FMH-7 it creates as a result of this APPCCMD that no error log data follows the FMH-7, and the AREA field in the RPL is ignored.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SENSE=*user-supplied_32-bit_fmh-7_ sense_code*

SENSE=(*user-supplied_32-bit_fmh-7_sense_ code_register***)**
Specifies the user-specified sense code that the application program requests to be placed in the FMH-7 that VTAM creates as a result of this APPCCMD macroinstruction. This sense code must be appropriate to the error. Otherwise, improper processing of the macroinstruction might result. This is the only one of the abnormal DEALLOC macroinstructions for which this field is applicable. This field is labeled RPL6SNSO in the RPL extension. For a list of valid sense codes, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG

X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is END_CONV after successful processing.

For full-duplex conversations, the conversation state is FDX_RESET after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID

RCPRI	RCSEC	Meaning
X'002C'	X'000B'	PARAMETER_ERROR—INCOMPLETE_GDS_VARIABLE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0021'	PARAMETER_ERROR—ABNORMAL_DEALLOCATE_ REJECTED_RETRY
X'002C'	X'002D'	PARAMETER_ERROR—INVALID_SENSE_CODE_ VALUE_SPECIFIED
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=DEALLOC, QUALIFY=CONFIRM

Purpose

This macroinstruction sends a confirmation request to a partner application program and, if the partner sends a positive confirmation response, VTAM deallocates the conversation.

Usage

QUALIFY=CONFIRM is used to ensure that the partner receives all data on a conversation before that conversation is deallocated.

VTAM sends the partner LU any remaining data in the SEND buffer, which is followed by a confirmation request. If the partner LU sends a positive response to the confirmation request, VTAM deallocates the conversation. If the partner LU sends a negative response to the confirmation request, VTAM does not deallocate the conversation. This macroinstruction completes only after a response is received from the partner LU. It corresponds to the DEALLOCATE (TYPE=CONFIRM) verb in the LU 6.2 architecture.

When this macroinstruction completes, the current conversation state is in the CONSTATE field.

Because this macroinstruction requests deallocation of the conversation, the data in the SEND buffer must complete a logical record.

For more information on sending and responding to confirmation requests, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction can only be used on half-duplex conversations from the SEND conversation state.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=CONFIRM (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►─────────────────────────────────────────────────────────────────────────────►
 ,AAREA= rpl_extension_address_field (1)
         (rpl_extension_address_register)
►─────────────────────────────────────────────────────────────────────────────►
 ,ACB= acb_address_field (1)          ,BRANCH= NO (1)
       (acb_address_register)                 YES
```

```
►──┬──────────────
```

associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register*)
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program

immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension. It can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5

has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
Indicates that an FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner application program. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Indicates either that no error indicator was received or that an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that VTAM did not recognize. This field is labeled RPL6SNSI in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

The conversation state is END_CONV after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_ CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID

RCPRI	RCSEC	Meaning
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_ SUPPORTED_BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_ NO_RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ NOT_VALID_FOR_FULL-DUPLEX_CONVERSATIONS
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=DEALLOC, QUALIFY=DATACON

Purpose

This macroinstruction sends data, which is supplied by the application program, and any information in the SEND buffer to a partner application program, followed by a confirmation request. If the partner LU sends a positive response to the confirmation request, VTAM deallocates the conversation normally.

Usage

This macroinstruction is used to send data to the partner LU and to ensure that the partner receives all the data before the conversation is deallocated.

VTAM sends any data remaining in the buffer followed by the data specified on the macroinstruction to the partner LU. This data is followed by a confirmation request. The macroinstruction completes only after the partner LU responds to the confirmation request. If the partner sends a positive confirmation response, the conversation is deallocated. If the partner LU sends a negative confirmation response, the conversation is not deallocated. This macroinstruction corresponds to the SEND_DATA and DEALLOCATE (TYPE=CONFIRM) verbs in the LU 6.2 architecture.

When this macroinstruction completes, the current conversation state is found in the CONSTATE field.

Because this macroinstruction requests deallocation of the conversation, the data sent must complete a logical record.

For more information on sending and responding to confirmation requests, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction can be used only on half-duplex conversations from the SEND conversation state.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=DATACON (2) ,RPL= [rpl_address_field (8)
                                                          (rpl_address_register)]
►
 [,AAREA= [rpl_extension_address_field (1)
           (rpl_extension_address_register)]]
►
 [,ACB= [acb_address_field (1)
         (acb_address_register)]]
```

```
►──┬──────────────
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

  If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs

running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type

macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.

- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

This field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
Indicates that an FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner application program. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Indicates either that no error indicator was received or that an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not recognized by VTAM. This field is labeled RPL6SNSI in the RPL extension.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. It is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether the application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). This field is labeled RPL6RSIG in the RPL extension.

YES (B'1')

Indicates that a SIGNAL RU has been received from the partner application program. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')

Indicates that no SIGNAL RU has been received from the partner application program. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF

The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS

The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

The conversation state is END_CONV after successful processing.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC

RCPRI	RCSEC	Meaning
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ NOT_VALID_FOR_FULL-DUPLEX_CONVERSATIONS
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=DEALLOC, QUALIFY=DATAFLU

Purpose

This macroinstruction unconditionally deallocates a conversation after sending data to a partner application program. The send function of the macroinstruction includes flushing the SEND buffer.

Usage

This macroinstruction combines the functions of two macroinstructions, APPCCMD CONTROL=SEND, QUALIFY=DATA followed by APPCCMD CONTROL=DEALLOC, QUALIFY=FLUSH. As with all macroinstructions that both send data and deallocate a conversation, the data sent by the application program *must complete a logical record*.

The deallocation request on this macroinstruction is unconditional. After VTAM successfully sends the data, it deallocates the conversation. Any incoming error information received for the application program is discarded.

This macroinstruction corresponds to the SEND_DATA verb followed by the DEALLOCATE (TYPE=FLUSH) verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the SEND or PENDING_SEND conversation states.

For full-duplex conversations, this macroinstruction can be issued from the following states:

- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
  [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=DATAFLU (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
► [ ,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1) ] ►
► [ ,ACB= [acb_address_field | (acb_address_register)] (1) ] ►
► [ ,AREA= [data_area_or_buffer_list_address_field | (data_area_or_buffer_list_address_register)] (1) ] [ ,BRANCH= [NO | YES] (1) ] ►
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                          │
   └─,CONMODE=──┬─BUFFCA──────────────┬──┬──────────────────────────────────────────────────────────────────────────────────────────┘
                │                     │  │                                                                                          │
                ├─CS──────────────────┤  └─,CONVID=──┬─32-bit_resource_id_field──────────────────────────────────────────────────────┘
                │                     │              │                                                                                │
                ├─LLCA────────────────┤              └─(32-bit_resource_id_register)──────────────────────────────────────────────┘
                │                     │
                └─SAME──────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                          │
   └─,CONXMOD=──┬─CA──────────────────┬──┬──────────────────────────────────────────────────────────────────────────────────────────┘
                │                     │  │                                                                                          │
                ├─CS──────────────────┤  └─,CRYPT=──┬─NO──────────────────────────────────────────────────────────────────────────┘
                │                     │             │                                                                               │
                └─SAME──────────────────┘             └─YES─────────────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (3)    (1)                                                                   │
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
   │        │                                                                                                                       │
   │        ├─ecb_address_field──────────────────────────────────────────────────────────────────────────────────────────────────┤
   │        │                                                                                                                       │
   │        └─(ecb_address_register)──────────────────────────────────────────────────────────────────────────────────────────────┤
   │                                                                                  (4)    (1)                                   │
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────────┘
             │                                                                                                                       │
             └─(exit_routine_address_register)──────────────────────────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                  (7)                                                                                          │
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────┬─)──┬──────────────────────────────────────────────────────────────────────────────
```

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

  If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.
- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension. For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
Returns the sense code carried in the FMH-7 used in deallocating the conversation. This field is labeled RPL6SNSI in the RPL extension.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. It is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). This field is labeled RPL6RSIG in the RPL extension.

YES (B'1')
Indicates that a SIGNAL RU has been received from the partner application program. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
Indicates that no SIGNAL RU has been received from the partner application program. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation enters END_CONV after successful completion of the macroinstruction.

For full-duplex conversations, the conversation enters one of the following states after successful completion of the macroinstruction.

- RECEIVE_ONLY
- PENDING_RECEIVE-ONLY_LOG
- FDX_RESET

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'0004'	X'0002'	CONVERSATION_TYPE_MISMATCH
X'0004'	X'0003'	PIP_NOT_ALLOWED
X'0004'	X'0004'	PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	SECURITY_NOT_VALID
X'0004'	X'0006'	SYNC_LEVEL_NOT_SUPPORTED_BY_LU
X'0004'	X'0007'	SYNC_LEVEL_NOT_SUPPORTED_BY_PROGRAM
X'0004'	X'0008'	TPN_NOT_RECOGNIZED
X'0004'	X'0009'	TRANSACTION_PROGRAM_NOT_AVAILABLE_NO_RETRY
X'0004'	X'000A'	TRANSACTION_PROGRAM_NOT_AVAILABLE_RETRY
X'0004'	X'000B'	CANNOT_RECONNECT_TRANSACTION_PROGRAM_NO_RETRY
X'0004'	X'000C'	CANNOT_RECONNECT_TRANSACTION_PROGRAM_RETRY
X'0004'	X'000D'	RECONNECT_NOT_SUPPORTED_BY_PROGRAM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT

RCPRI	RCSEC	Meaning
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNCATION
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNCATING
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNCATION
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	FOLLOWING_NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00AC'	X'0001'	ERROR_INDICATION_RECEIVED_DEALLOCATE ABEND PROGRAM
X'00AC'	X'0002'	ERROR_INDICATION_RECEIVED_DEALLOCATE ABEND SERVICE
X'00AC'	X'0003'	ERROR_INDICATION_RECEIVED_DEALLOCATE ABEND TIME
X'00AC'	X'0004'	ERROR_INDICATION_RECEIVED_ALLOCATION ERROR
X'00AC'	X'0005'	ERROR_INDICATION_RECEIVED_UNKNOWN ERROR CODE
X'00AC'	X'0006'	ERROR_INDICATION_RECEIVED_RESOURCE FAILURE, RETRY
X'00AC'	X'0007'	ERROR_INDICATION_RECEIVED_RESOURCE FAILURE, NO RETRY
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=DEALLOC, QUALIFY=FLUSH

Purpose

This macroinstruction flushes the SEND buffer and unconditionally deallocates a conversation.

Usage

For half-duplex conversations, this macroinstruction executes the function of the APPCCMD CONTROL=SEND, QUALIFY=FLUSH macroinstruction prior to the deallocation. Any error information coming from the partner application program that is received by VTAM after the macroinstruction is issued is not reported to the application program.

This macroinstruction, when issued on a full-duplex conversation, either initiates the conversation deallocation or completes the conversation deallocation if a deallocation request has been received from the conversation partner.

This macroinstruction corresponds to the DEALLOCATE (TYPE=FLUSH) verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from a SEND or PENDING_SEND conversation state.

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:

- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=DEALLOC ,QUALIFY=FLUSH (2) ,RPL= [rpl_address_field (8)
                                                         (rpl_address_register)]
►
 [,AAREA= [rpl_extension_address_field (1)
           (rpl_extension_address_register)]]
►
 [,ACB= [acb_address_field (1)
         (acb_address_register)]]  [,BRANCH= [NO (1)
                                              YES]]
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                
```

associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register*)
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program

immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation enters END_CONV after successful completion of the macroinstruction.

For full-duplex conversations, the conversation can enter the following conversation states after successful processing:

- RECEIVE_ONLY
- PENDING_RECEIVE-ONLY_LOG
- FDX_RESET

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00AC'	X'0001'	ERROR_INDICATION_RECEIVED_DEALLOCATE_ ABEND_PROGRAM
X'00AC'	X'0002'	ERROR_INDICATION_RECEIVED_DEALLOCATE_ ABEND_SERVICE
X'00AC'	X'0003'	ERROR_INDICATION_RECEIVED_DEALLOCATE_ ABEND_TIME
X'00AC'	X'0004'	ERROR_INDICATION_RECEIVED_ALLOCATION_ERROR
X'00AC'	X'0005'	ERROR_INDICATION_RECEIVED_UNKNOWN_ERROR_CODE
X'00AC'	X'0006'	ERROR_INDICATION_RECEIVED_RESOURCE_ FAILURE_RETRY
X'00AC'	X'0007'	ERROR_INDICATION_RECEIVED_RESOURCE_ FAILURE_NO_RETRY

APPCCMD CONTROL=DEALLOCQ

Purpose

This macroinstruction deallocates a conversation when an application program has detected an error. This macroinstruction is queued if the conversation is in the RECEIVE state and has not yet received data. When data is received, VTAM continues deallocation of the conversation.

Usage

QUALIFY=ABNDPROG is used to abnormally terminate a conversation when the application program detects an error that will prevent further useful conversation.

QUALIFY=ABNDSERV is used to abnormally terminate a conversation and alert VTAM that an LU service component has encountered an error.

QUALIFY=ABNDTIME is used to abnormally terminate a conversation when the application program detects that it has not received information from its partner for a specified amount of time.

QUALIFY=ABNDUSER is used to abnormally terminate a conversation. The command also alerts VTAM that the application program will provide a user-specified sense code to place in the FMH-7 that VTAM creates as a result of this command. The application program is responsible for the validity of the sense code.

This macroinstruction abnormally deallocates a conversation. If the conversation is in a sending state, the function is identical to the abnormal termination APPCCMD CONTROL=DEALLOC. The SEND buffer is flushed before the conversation is deallocated.

If the conversation is in a receiving state and is waiting for a first, or only element in the chain, this macroinstruction is queued until data is received from the partner LU.

To contrast this macroinstruction with DEALLOC, the DEALLOCQ macroinstruction will never receive an RCPRI, RCSEC of X'002C', X'0021'.

The following macroinstructions cannot be canceled by APPCCMD CONTROL=DEALLOCQ:

- APPCCMD CONTROL=RECEIVE, QUALIFY=ANY that has not been matched to a conversation
- APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY that has not been matched to a conversation
- APPCCMD CONTROL=RCVFMH5, QUALIFY=NULL|QUEUE
- APPCCMD CONTROL=RESETRCV
- APPCCMD CONTROL=OPRCNTL
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDPROG|ABNDSERV|ABNDTIME|ABNDUSER
- APPCCMD CONTROL=DEALLOCQ, QUALIFY=ABNDPROG|ABNDSERV|ABNDTIME|ABNDUSER
- APPCCMD CONTROL=TESTSTAT, QUALIFY=ALL|IALL
- A macroinstruction that is waiting for a response to a confirmation request
- A macroinstruction that is waiting for the arrival of an FMH-7

If any one of these macroinstructions is outstanding, the application program can either wait for the outstanding APPCCMD to complete and then issue APPCCMD CONTROL=DEALLOCQ or issue APPCCMD CONTROL=REJECT.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for information on deallocating a conversation when an error is detected.

For early deallocation of a pending APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5, QUALIFY=DATAQUE, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- PENDING_ALLOCATE
- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PENDING_SEND
- PENDING_END_CONV_LOG
- PENDING_RECEIVE_LOG

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:

- PENDING_ALLOCATE
- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=DEALLOCQ ,QUALIFY= ABNDPROG (2) ,RPL= rpl_address_field (8) ►
                                           ABNDSERV
                                           ABNDTIME
                                           ABNDUSER
                                           (rpl_address_register)

► [,AAREA= rpl_extension_address_field (1)] [,ACB= acb_address_field (1)] ►
          (rpl_extension_address_register)        (acb_address_register)

► [,AREA= data_area_or_buffer_list_address_field (1)] [,BRANCH= NO (1)] ►
          (data_area_or_buffer_list_address_register)          YES
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                   (2)
   └─,CONVID=──┬─32-bit_resource_id_field──────────────┬──────────────────────────────────────────────────────────────────────
               └─(32-bit_resource_id_register)─────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                   (3)   (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────┬──────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field─────────────────┤
   │        └─(ecb_address_register)────────────┘
   │                                                   (4)   (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────┬──────────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)─────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                  ┌──────────────────────────────┐
   │                  ▼,                             │
   │                    (7)                          │
   └─,OPTCD=(──┬──────────────────────────────────────┬──)──────────────────────────────────────────────────────────────────
               │          (1)                         │
               ├──┬─ASY──────────────────────────────┤
               │  │          (1)                     │
               │  └─SYN──────────────────────────────┤
               │                (1)   (5)             │
               ├──┬─KEEPSRB──────────────────────────┤
               │  │                (1)   (6)         │
               │  └─NKEEPSRB──────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                   (1)
   └─,RECLEN=──┬─optional_log_data_length──────────────┬──────────────────────────────────────────────────────────────────────
               └─(optional_log_data_length_register)───┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                   (2)
   └─,SENSE=──┬─32-bit_unbind_sense_code──────────────┬──────────────────────────────────────────────────────────────────────
              └─(32-bit_unbind_sense_code_register)───┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*optional_log_data_area_address_field*

AREA=(*optional_log_data_area_address_register***)**
Specifies the address of a data area containing a formatted error log GDS variable to be sent to the partner application program. The application program is responsible for placing the error log data into the local system log. If the application program chooses to supply an error log GDS variable, it has to supply the entire GDS variable on the APPCCMD macroinstruction. VTAM inspects the 2-byte logical-record length (LL) field of the GDS variable to determine if the amount of data supplied is equal to the length specified in the LL field. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the error log GDS variable.) This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*optional_log_data_length*

RECLEN=(*optional_log_data_length_register***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL. A value of 0 in the RECLEN field indicates that the application program has chosen not to provide optional error log data to VTAM. If the application program specifies RECLEN=0, VTAM indicates in the FMH-7 it creates as a result of this APPCCMD that no error log data follows the FMH-7, and the AREA field in the RPL is ignored.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SENSE
Specifies the user-specified sense code that the application program requests to be placed in the FMH-7 that VTAM creates as a result of this APPCCMD macroinstruction. This sense code must be appropriate to the error. Otherwise, improper processing of the macroinstruction might result. This field is examined

only if QUALIFY=ABNDUSER is issued. This field is labeled RPL6SNSI in the RPL extension. For a list of valid sense codes,

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an

FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, END_CONV state is entered.

For full-duplex conversations, FDX_RESET state is entered.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (DEALLOCATION IS COMPLETE)
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000B'	PARAMETER_ERROR—INCOMPLETE_GDS_VARIABLE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002D'	PARAMETER_ERROR—INVALID_SENSE_CODE_ VALUE_SPECIFIED
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA.
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS

Purpose

This macroinstruction responds positively to a session establishment request.

Usage

This macroinstruction is issued after the application program is notified through its LOGON or SCIP exit routine that a CINIT or BIND request has been received. (For a description of when the LOGON and SCIP exits are scheduled and for the information provided in each exit, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.) The function of this command is similar to the VTAM API commands OPNDST OPTCD=ACCEPT and OPNSEC for non-LU 6.2 sessions.

When this macroinstruction is used in a LOGON exit, the RPLAREA field of the read-only RPL passed to the exit routine contains the address of a read-only copy of the CINIT. The application program can examine the parameters of the BIND in the CINIT. If the application program needs to override any of the BIND parameters, it can specify session parameters for a BIND on this macroinstruction (mapped by ISTDBIND).

Attention: If both the local and the partner LU are the same LU, then this macroinstruction must not be issued from the LOGON exit routine. Otherwise, the session will hang.

The partner LU can negotiate the BIND. If this occurs, VTAM verifies and accepts the negotiated BIND parameters. (For information on BIND fields and their settings, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.) However, VTAM does not return the negotiated BIND response to the application program when this macroinstruction completes.

The application program can use this macroinstruction in a SCIP exit to override some of the values received in the BIND by providing a BIND image (in ISTDBIND format) to be used in building a response. When this macroinstruction is used in a SCIP exit, word 4 of the parameter list points to session parameters mapped by ISTDBIND. If the application program needs to override any of the BIND parameters, it can specify session parameters for a BIND response on this macroinstruction (mapped by ISTDBIND). Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for details on the values that can be overridden.

Note: APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS does not correspond to the ACTIVATE_SESSION verb described in the LU 6.2 architecture.

Context

Input states are not applicable to this macroinstruction.

Syntax

```
►►                                                  (2)                                  (8)
──┬──────────┬──APPCCMD CONTROL=OPRCNTL──,QUALIFY=ACTSESS──────────,RPL=──┬─rpl_address_field──────────┬──►
  └─name─────┘                                                           └─(rpl_address_register)────┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,AAREA=──┬─rpl_extension_address_field──────────────────────────────────────────────────────────────────────────┘
             │                                                                                   (1)
             └─(rpl_extension_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,ACB=──┬─acb_address_field──────────────────────────────────────────────────────────────────────────────────────┘
           │                                                                                   (1)
           └─(acb_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,AREA=──┬─0──────────────────────────────────────────────────────────────────────────────────────────────────┬┘
            │                                                                                                    (1)
            ├─session_parameter_address_field──────────────────────────────────────────────────────────────────┤
            └─(session_parameter_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,ARG=──┬─4-byte_session_identifier_(cid)_field──────────────────────────────────────────────────────────────┬┘
           │                                                                                                    (1)
           └─(4-byte_session_identifier_(cid)_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,BRANCH=──┬─NO──┬┘
              │     │                                                                                            (1)
              └─YES─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,CONFTXT=──┬─NO──┬┘
               │     │                                                                                            (1)
               └─YES─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────┬┘
           │                                                                                                (3)  (1)
           ├─ecb_address_figld──────────────────────────────────────────────────────────────────────────────────┤
           └─(ecb_address_register)─┘
                                                                                                                  (4)  (1)
           ,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────┬┘
                   │                                                                                            (1)
                   └─(exit_routine_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   └,OPTCD=(──┬──┬──┬
```

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPl or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*session_parameter_address_field*

AREA=(*session_parameter_address_register***)**
Specifies the address of an area that contains a set of session parameters that VTAM uses when constructing the BIND or BIND response, which is sent to establish a session. If an address is indicated, the set of parameters specified by the application program will override the session parameters given in the CINIT or BIND (refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for information on building the session parameters). This field is labeled RPLAREA in the RPL. If you specify AREA=0, VTAM uses the set of session parameters contained in the CINIT or BIND to construct the BIND or BIND response.

Note: You should use the ISTDBIND DSECT if you include user data fields on the BIND.

ARG=*4-byte_session_identifier_(cid)_field*

ARG=(*4-byte_session_identifier_(cid)_register***)**
Specifies the CID of the session that was returned to the application program in the parameter list of the LOGON or SCIP exit routine. The specified CID must identify a CINIT or BIND that is queued for this application program.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONFTXT
Indicates whether or not data sent or received on this session is to be considered "confidential" within this host. This field is labeled RPL6CFTX in the RPL extension.

CONFTXT=YES
The VTAM buffers used to hold the data are cleared before they are returned to their buffer pools.

CONFTXT=NO
No clearing is performed.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the

completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*session_parameter_information_length*

RECLEN=(*session_parameter_information_length_register***)**
Specifies the length of the session parameter information. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of information that the application program can associate with this operator control request. The information is returned unchanged when the macroinstruction completes. This data cannot be used by any conversations. It can be used for correlation purposes. This field is labeled RPL6USR in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
Contains the sense code if any is returned from session initiation macroinstructions. This field is labeled RPL6SNSI in the RPL extension.

USERFLD
Returns any unchanged user data that the application program placed in this field. This field is labeled RPL6USR in the RPL extension.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0009'	PARAMETER_ERROR—INCOMPLETE_STRUCTURE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0014'	PARAMETER_ERROR—INVALID_BIND_PARAMETERS
X'002C'	X'001E'	PARAMETER_ERROR—CID_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON_APPC
X'0064'	X'0000'	ACTIVATION_FAILURE
X'0068'	X'0000'	LU_MODE_SESSION_LIMIT_EXCEEDED
X'006C'	X'0000'	SESSION_NOT_PENDING
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS

Purpose

This macroinstruction negotiates the session limits on a mode group between the application program and a partner application.

Usage

VTAM determines the new session limits by using the session limits specified on the macroinstruction and the defined session limits of the partner LU. The overall session limits, the contention-winner session limits, and the contention-loser session limits are negotiated. Other parameters, such as draining of a conversation request and responsibility for deactivation, are also negotiated by this macroinstruction.

When this macroinstruction completes, VTAM can activate or deactivate sessions to make them conform to the new session limits. However, sessions already assigned to a conversation are not deactivated.

This macroinstruction corresponds to the INITIALIZE_SESSION_LIMIT, CHANGE_SESSION_LIMIT, and RESET_SESSION_LIMIT verbs in the LU 6.2 architecture.

The APPCCMD CONTROL=OPRCNTL, QUALIFY=DEFINE macroinstruction can be used by a partner LU that is capable of parallel sessions to define the session limits that can be used in the negotiation when it receives the CNOS request.

For a full discussion of this macroinstruction, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

Input states are not applicable to this macroinstruction.

When a mode is retained for persistent LU-LU sessions, the QUALIFY=CNOS macroinstruction is not allowed.

Syntax

```
►►                                                  (2)                                  (8)
 ┌──────────┐APPCCMD CONTROL=OPRCNTL──,QUALIFY=CNOS──────────,RPL=┬─rpl_address_field──────┬──►
 └─name─────┘                                                    └─(rpl_address_register)─┘

►──┬──────────────
```

```
►──┬──────────────
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*cnos_session_limits_structure_address_field*

AREA=(*cnos_session_limits_structure_address_register***)**
Specifies the address of a data area containing a CNOS session limits data structure. (See "CNOS Session Limits Data Structure (ISTSLCNS)" on page 645 for the VTAM-supplied DSECT that can be used to fill in and test values.) The specification of a session limits structure is optional (the AREA field in the RPL extension would be 0 in this case). The defaults that are used when a session limits structure is omitted are given in the description of each parameter. The fields in the data structure that apply to this macroinstruction are described in the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***) This field is labeled RPLEXTDS in the RPL.**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name that requires the session limit and contention-winner polarity values to be changed. The mode name specified can be any mode name that is valid as the LOGMODE value on the APPCCMD CONTROL=ALLOC macroinstruction including the SNASVCMG mode name, which is used for exchanging the CNOS request and reply when the application program and partner application are connected by parallel sessions. However, no CNOS flow occurs to the partner application program as a result of issuing this macroinstruction for the SNASVCMG mode name.

If the session limits control block specifies that SESSLIM=0 and NBRMODE=ALL, the session limit negotiation applies to all noncontrol modes between the two LUs, and this parameter is ignored.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is less than eight characters, VTAM pads it on the right with blanks. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information). This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program to which the change in the session limit and contention-winner polarity values applies. The LU name is a name that is valid as the LU name value on the APPCCMD CONTROL=ALLOC macroinstruction.

The LU name can be up to eight characters in length. If it is less than eight characters, VTAM pads it on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NAMEUSE
Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME
The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME
The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program to which the change in the session limit and contention-winner polarity value applies.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.) The network identifier is also used to verify and update the logon mode table. It is the same as the NETID value on the APPCCMD CONTROL=ALLOC macroinstruction.

The network identifier can be up to eight characters in length. If it is fewer than eight characters, VTAM pads it on the right with blanks. This field is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*cnos_session_limits_structure_length*

RECLEN=(*cnos_session_limits_structure_length_register***)**
Specifies the length of the CNOS session limits data structure supplied by the AREA field. The application program must supply the entire session limits data structure; it cannot supply a partial structure. This field is only applicable if a CNOS session limits structure is specified by the AREA field. Otherwise, it is ignored by VTAM. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of information that the application program can associate with this operator control request. The information is returned unchanged when the macroinstruction completes. This data cannot be used by any conversations. It can be used for correlation purposes. This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register*)
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register*)
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This sense code is returned for the control operator session that VTAM establishes as part of processing the CNOS request. This field is labeled RPL6SNSI in the RPL extension.

USERFLD
Returns any unchanged user data that the application program placed in this field. This field is labeled RPL6USR in the RPL extension.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Name change vector (X'18')
- Partner's application capabilities vector (X'1A')

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0001'	OK—AS_SPECIFIED
X'0000'	X'0002'	OK—AS_NEGOTIATED
X'0000'	X'000C'	OK_AS_SPECIFIED—PARTNER_LU_KNOWN_ BY_DIFFERENT_NAME
X'0000'	X'000D'	OK_AS_NEGOTIATED—PARTNER_LU_KNOWN_ BY_DIFFERENT_NAME
X'0000'	X'0004'	OK—PARTNER LU SUPPORTS SINGLE SESSION
X'0008'	X'0000'	CNOS_ALLOCATION_ERROR—ALLOCATION_FAILURE_ NO_RETRY
X'0008'	X'0001'	CNOS_ALLOCATION_ERROR—ALLOCATION_FAILURE_ RETRY
X'0008'	X'0002'	CNOS_ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_ RETRY
X'0008'	X'0003'	CNOS_ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_ NO_RETRY

RCPRI	RCSEC	Meaning
X'0008'	X'0004'	CNOS_ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0008'	X'0005'	CNOS_ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0008'	X'0006'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0008'	X'0007'	NETWORK-QUALIFIED_NAME_MISMATCH
X'000C'	X'0000'	CNOS_RESOURCE_FAILURE_NO_RETRY
X'0010'	X'0000'	COMMAND_RACE_REJECT—PARTNER_GRANTED_RETRY
X'0010'	X'0001'	COMMAND_RACE_REJECT—COPR_FOR_LOCAL_LU_ RETRIED
X'0010'	X'0002'	COMMAND_RACE_REJECT—PARTNER_CNOS_IN_ PROGRESS
X'0010'	X'0003'	COMMAND_RACE_REJECT—LU_IS_IN_PENDING_ SINGLE_STATE
X'0010'	X'0004'	COMMAND_RACE_REJECT—PARTNER_LU_STARTING_ SESSION
X'0020'	X'0000'	CNOS_FAILURE_RETRY
X'0028'	X'0000'	LU_MODE_SESSION_LIMIT_CLOSED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'0004'	PARAMETER_ERROR—INVALID_VALUES_FOR_ SNASVCMG_MODE
X'002C'	X'0005'	PARAMETER_ERROR—INVALID_DRAINL_CHANGE
X'002C'	X'0006'	PARAMETER_ERROR—SNASVCMG_MODE_CANNOT_ CURRENTLY_BE_RESET
X'002C'	X'0007'	PARAMETER_ERROR—MINWINL_PLUS_MINWINR_ EXCEEDS_SESSLIM
X'002C'	X'0008'	SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'0009'	PARAMETER_ERROR—INCOMPLETE_STRUCTURE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0018'	PARAMETER_ERROR—INVALID_LIMIT_SPECIFIED
X'002C'	X'0019'	PARAMETER_ERROR—SNASVCMG_MODE_ALREADY_ INITIALIZED
X'002C'	X'001A'	PARAMETER_ERROR—ALL_MODES_SPECIFIED_ON_ SINGLE_SESSION_LU
X'002C'	X'001B'	PARAMETER_ERROR—SNASVCMG_OR_CPSVCMG_ MODE_FOR_SINGLE_SESSION_LU
X'002C'	X'001C'	PARAMETER_ERROR—SINGLE_SESSION,_MODE_ ALREADY_INITIALIZED
X'002C'	X'001E'	CID_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0054'	X'0000'	UNRECOGNIZED_MODE_NAME
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB

RCPRI	RCSEC	Meaning
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY

APPCCMD CONTROL=OPRCNTL, QUALIFY=DACTSESS

Purpose

This macroinstruction responds negatively to a request for session establishment.

Usage

This command is issued after the application program is notified through its LOGON or SCIP exit routine that a CINIT or BIND request has been received. The function of this command is similar to the VTAM API commands CLSDST RELEASE and SESSIONC CONTROL=BIND for non-LU 6.2 sessions.

When this macroinstruction is used in a LOGON exit, the RPLAREA field of the read-only RPL contains a read-only copy of the CINIT. After examining the BIND image in the CINIT, the application program can issue this macroinstruction to prevent the session from being activated.

When this macroinstruction is used in a SCIP exit, the RPLAREA field of the read-only RPL contains the address of a read-only copy of the BIND. After examining the BIND, the application program can issue this macroinstruction to prevent the session from being activated.

APPCCMD CONTROL=OPRCNTL, QUALIFY=DACTSESS does not correspond to the DEACTIVATE_SESSION verb described in the LU 6.2 architecture.

Context

Input states are not applicable to this macroinstruction.

Syntax

```
►►
  [name] APPCCMD CONTROL=OPRCNTL ,QUALIFY=DACTSESS (2) ,RPL= rpl_address_field (8)
                                                            (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_figure (1)
        (acb_address_register)
►
  ,ARG= 4-byte_session_identifier_(cid)_field (1)
        (4-byte_session_identifier_(cid)_register)
  ,BRANCH= NO (1)
           YES
►
  ,ECB= INTERNAL (3) (1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field (4) (1)
         (exit_routine_address_register)
```

```
►──┬──────────────
```

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM

returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of information that the application program can associate with this operator control request. The information is returned unchanged when the macroinstruction completes. This data cannot be used by any conversations. It can be used for correlation purposes. This field is labeled RPL6USR in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Returns any unchanged user data that the application program placed in this field. This field is labeled RPL6USR in the RPL extension.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD

RCPRI	RCSEC	Meaning
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001E'	PARAMETER_ERROR—CID_INVALID
X'006C'	X'0000'	SESSION_NOT_PENDING
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=OPRCNTL, QUALIFY=DEFINE

Purpose

This macroinstruction changes the session limit values that have been defined and that are used to negotiate a CNOS request from a partner LU. It also displays selected fields from the LU-mode table in the DEFINE/DISPLAY (ISTSLD) control block.

Usage

This macroinstruction can be used to modify values in a mode name entry that were originally obtained by VTAM from the APPL definition statement or that were supplied by using this macroinstruction previously. There is no direct correlation to the DEFINE verb in the LU 6.2 architecture.

The session limit values that are defined are passed to VTAM in a DEFINE/DISPLAY control block. You must specify the address of this control block in the RPL when issuing the macroinstruction. The address is contained in the RPLAREA field, which can be set with the AREA keyword.

Most of the values specified in the DEFINE/DISPLAY control block are used to negotiate the values received in a CNOS request sent by the partner application program. The values are not affected by, nor do they have any effect upon, the values specified through the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction. For example, an application program can define the session limit used for negotiating purposes to be 10, yet later issue a CNOS macroinstruction that specifies a session limit of 20. The defined value of 10 does not restrict the CNOS value of 20; the CNOS value of 20 does not cause the defined value of 10 to be changed.

When this macroinstruction is issued before a CNOS request is negotiated on a mode, VTAM creates an entry in the LU-mode table for the mode and places the defined session limits in the table. The negotiated session limits are not determined until a CNOS request is negotiated.

When issuing APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS, the application program can elect not to specify the limits to be used for CNOS negotiation. If this occurs, VTAM uses the defined limits specified by this macroinstruction as the default for these values. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

This macroinstruction can also be used to help control VTAM's use of storage. Specifying default limits of 0 with DELETE=ALLOW (in the DEFINE/DISPLAY session limits control block) informs VTAM that this mode name can be deleted from the LU-mode table when the mode name is no longer being used. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for an example of setting the DEFINE/DISPLAY session limits control block.) The execution of this macroinstruction involves only the application program; it does not cause any information to be sent through the network. The specified field values are in effect once the execution completes.

Context

Input states are not applicable to this macroinstruction.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
  [name] APPCCMD CONTROL=OPRCNTL ,QUALIFY=DEFINE (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_field (1)
        (acb_address_register)
►
  ,AREA= define/display_session_limits_structure_address_field (1)
         (define/display_session_limits_structure_address_register)
►
  ,BRANCH= NO (1)
           YES
  ,ECB= INTERNAL (3) (1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field (4) (1)
         (exit_routine_address_register)
►
  ,LOGMODE=8-byte_logon_mode_name (2)
  ,LUNAME=8-byte_lu_name (2)
►
  ,NETID=8-byte_network_identifier (2)
  ,OPTCD=( ,
           (7)
           ASY (1)
           SYN (1)
           KEEPSRB (1) (5)
           NKEEPSRB (1) (6) )
►
  ,RECLEN= define/display_session_limits_structure_length (1)
           (define/display_session_limits_structure_length_register)
►
  ,USERFLD= 4-bytes_of_user_data (2)
            (user_data_register)
►◄
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*define/display_session_limits_structure_address_field*

AREA=(*define/display_session_limits_structure_address_register***)**
Specifies the address of a data area containing a DEFINE/DISPLAY session limits data structure. (See "DEFINE/DISPLAY Session Limits Data Structure (ISTSLD)" on page 646 for a description of the IBM-supplied DSECT that can be used to map this storage.) A description of the fields in the control block can be found in *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program to which the change in the session limit and contention-winner polarity values applies. This LU name is

the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program to which the change in the session limit and contention-winner polarity value applies.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.) The network identifier also is used to find and update the contents of the logon mode table.

This network identifier is an identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*define/display_session_limits_structure_length*

RECLEN=(*define/display_session_limits_structure_register***)**
Specifies the length of the DEFINE/DISPLAY session limits data structure supplied by the AREA field. The application program must supply the entire session limits data structure; it cannot supply a partial structure. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register*)
Specifies 4 bytes of information that the application program can associate with this operator control request. The information is returned unchanged when the macroinstruction completes. This data cannot be used by any conversations. It can be used for correlation purposes. This field is labeled RPL6USR in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Returns any unchanged user data that the application program placed in this field. This field is labeled RPL6USR in the RPL extension.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'0007'	PARAMETER_ERROR—MINWINL_PLUS_MINWINR_ EXCEEDS_SESSLIM
X'002C'	X'0009'	PARAMETER_ERROR—INCOMPLETE_STRUCTURE_ SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'0017'	PARAMETER_ERROR—INVALID_MODE_SPECIFIED
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED

RCPRI	RCSEC	Meaning
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A4'	X'0000'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_VARIANT_NAME_ENTRY
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_UNUSABLE_NAME_ENTRY

APPCCMD CONTROL=OPRCNTL, QUALIFY=DISPLAY

Purpose

This macroinstruction returns information associated with an LU or a mode name of an LU.

Usage

The information returned from this macroinstruction is contained in the DEFINE/DISPLAY control block. You must specify the address of this control block in the RPL when issuing the macroinstruction. It is contained in the RPLAREA field, which can be set with the AREA keyword. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the control block.

The execution of this macroinstruction involves only the application program. It does not cause any information to be sent through the network. There is no direct correlation to the DISPLAY verb described in the LU 6.2 architecture.

Context

Conversation states are not applicable to this macroinstruction.

Syntax

```
►►
  [name] APPCCMD CONTROL=OPRCNTL ,QUALIFY=DISPLAY (2) ,RPL= [rpl_address_field (8)
                                                            (rpl_address_register)]
►
  [,AAREA= [rpl_extension_address_field (1)
            (rpl_extension_address_register)]]
►
  [,ACB= [acb_address_field (1)
          (acb_address_register)]]
►
  [,AREA= [define/display_session_limits_structure_address_field (1)
           (define/display_session_limits_structure_address_register)]]
►
  [,AREALEN= [define/display_session_limits_structure_length (1)
              (define/display_session_limits_structure_length_register)]]
►
  [,BRANCH= [NO (1)
             YES]]
  [,ECB= [INTERNAL (3) (1)
          ecb_address_field
          (ecb_address_register)]
  [,EXIT= [exit_routine_address_field (4) (1)
           (exit_routine_address_register)]]
```

```
►──┬──────────────
```

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*define/display_session_limits_structure_address_field*

AREA=(*define/display_session_limits_structure_address_register***)**
Specifies the address of a data area for the DEFINE/DISPLAY session limits data structure (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of this control block.). This field is labeled RPLAREA in the RPL.

AREALEN=*define/display_session_limits_structure_length*

AREALEN=(*define/display_session_limits_structure_length_register***)**
Specifies the length of the area in which the DEFINE/DISPLAY session limits data structure is to be returned. If a mode name is specified for the LOGMODE field, the application program must supply an area large enough to contain the entire session limits data structure. If LOGMODE=0 is specified, a length of 40 can be coded for this field. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be allocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program to which the requested session information applies. The LU name is a name that is valid as the LU name value on the APPCCMD CONTROL=ALLOC macroinstruction and the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads it on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program to which the requested session information applies.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.) The network identifier also is used to find and update the contents of the logon mode table. It is the same as the NETID value on the APPCCMD CONTROL=ALLOC macroinstruction.

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of information that the application program can associate with this operator control request. The information is returned unchanged when the macroinstruction completes. This data cannot be used by any conversations. It can be used for correlation purposes. This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual length of the session limits structure being returned by the AREA field. If the application program specified LOGMODE=0, the value 40 is returned for this field. This field is labeled RPLRLEN in the RPL.

A description of the session limits structure is found in the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Returns any unchanged user data that the application program placed in this field. This field is labeled RPL6USR in the RPL extension.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Partner's application capabilities vector (X'1A')

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0013'	PARAMETER_ERROR—NO_CORRESPONDING_MODE_IN_ LOGMODE_TABLE

RCPRI	RCSEC	Meaning
X'002C'	X'0016'	PARAMETER_ERROR—NO_CORRESPONDING_LU_IN_ LOGMODE_TABLE
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE

Purpose

This macroinstruction is used to restore modes and their associated persistent LU-LU sessions that are pending recovery.

Usage

This macroinstruction can be used by an application program to restore modes and associated persistent LU-LU sessions that are pending recovery. A mode is restored only after any sessions for the mode are restored. A mode without sessions also must be restored.

A single LU-mode can be restored when the LU name and logon mode are specified on the RESTORE command. All modes for a specific LU are restored when only the LU name is specified. If neither the LU name (with its NETID, if applicable) nor the logon mode is specified, all LUs and modes in the LU-mode table are restored.

The application program specifies the amount of information that is to be returned in the RESTORE control block. To do this, it uses the LIST keyword in the RESTORE macroinstruction. The application program can specify LU-mode table information, LU-mode table and session information, or no information. If the application program requests information to be returned, it must specify the address of a data area to contain that information. The application program must provide the storage area in addition to specifying the address of the storage. This address is contained in the RPLAREA field, which can be set with the AREA keyword.

When the area pointed to by RPLAREA is large enough, the macroinstruction builds multiple RESTORE blocks in this area, if necessary. The RESTORE structures are placed in the area specified until the area is filled or the command is completed, whichever comes first.

For more information about the restore process, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*. For information about the RESTORE control block, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*. For an example of retrieving information that is returned, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

Input states are not applicable to this macroinstruction.

The recovering VTAM application program can issue this macroinstruction only after it issues the SETLOGON START macroinstruction. Otherwise, this macroinstruction is rejected.

Syntax

```
►►
┌──────────┐
└name      ┘──APPCCMD CONTROL=OPRCNTL──,QUALIFY=RESTORE──────────,RPL=──┬─rpl_address_field──────────┬──►
                                                  (2)                     │ (8)                        │
                                                                          └─(rpl_address_register)─────┘
```

```
►──┬──────────────
```

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*restore_structure_address_field*

AREA=(*restore_structure_address_register***)**
Specifies the address of a data area that returns one or more RESTORE data structures. It is used only with LIST=ALL or LIST=NOSESS. It is unnecessary when LIST=NONE is specified. This field is labeled RPLAREA in the RPL.

AREALEN=*restore_structure_length*

AREALEN=(*restore_structure_length_register***)**
Specifies the length of the area in which the RESTORE data structure is to be returned. It is used only with LIST=ALL or LIST=NOSESS. It is unnecessary when LIST=NONE is specified. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be

informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LIST
Specifies the information to be returned in the RESTORE structure, which describes the LUs, modes, and sessions that have been restored. This field is labeled RPL6LIST in the RPL extension.

LIST=ALL
Specifies that all LU, mode, and session information is returned in the RESTORE structure.

LIST=NONE
Specifies that no RESTORE structure is returned.

LIST=NOSESS
Specifies that all LU and mode information but no session information is returned in the RESTORE structure.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name which should be restored. The application program can specify a logon mode name with an LU name to give greater granularity over the scope of the command. LOGMODE can be specified only with LUNAME. The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is less than eight characters, VTAM pads it on the right with blanks. If this operand is coded on this macroinstruction and on the RPL extension, VTAM uses the specifications from the macroinstruction. This field is labeled RPL6MODE in the RPL extension.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner whose modes must be restored. It is the same as the LU name value on the APPCCMD CONTROL=ALLOC macroinstruction. It is also the network name of the target LU. When the application program does not specify the LU name, all LUs and modes are restored. Otherwise, only the modes associated with a specified LU name are restored. The LU name can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner whose modes must be restored.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored.

If NQNAME=YES is specified, LUNAME and NETID together form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.) The network identifier also is used to find the appropriate information on sessions and modes waiting to be restored. It is the same as the NETID value on the APPCCMD CONTROL=ALLOC macroinstruction.

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of information that the application program can associate with this operator control request. The information is returned unchanged when the macroinstruction completes. This data cannot be used by any conversations. It can be used for correlation purposes. This field is labeled RPL6USR in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the length of AREA used to contain the RESTORE structure(s) returned by the AREA field. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Returns any unchanged user data that the application program placed in this field. This field is labeled RPL6USR in the RPL extension.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (RESTORE complete.)
X'0000'	X'0006'	RESTORE_UNNECESSARY—NO_SESSIONS_TO_RESTORE
X'0000'	X'0007'	RESTORE_INCOMPLETE—INPUT_WORK_AREA_TOO_ SMALL
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0013'	NO_CORRESPONDING_MODE_IN_LM_TABLE
X'002C'	X'0016'	NO_CORRESPONDING_LU_IN_LM_TABLE
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0029'	INVALID_LIST_VALUE_SPECIFIED_ON_APPCCMD_FOR_ RESTORE
X'002C'	X'002B'	NETWORK-QUALIFIED_NAME_REQUIRED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'009C'	X'0001'	RESTORE_REJECTED—RESTORE_ISSUED_BEFORE_ SETLOGON_START
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

RCPRI	RCSEC	Meaning
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=PREALLOC, QUALIFY=ALLOCD

Purpose

This macroinstruction reserves a session without establishing a conversation. If a session is not available and session limits allow, VTAM activates a session for the conversation, if possible. Session related information can be passed from VTAM to the application before the application sends the FMH-5. The conversation is not active until the application issues the APPCCMD CONTROL=SENDFMH5 macroinstruction.

Usage

QUALIFY=ALLOCD is used when an application program preallocates a conversation and wants VTAM to queue the request if the request cannot be met immediately. This macroinstruction completes when VTAM reserves a session for a conversation or when an error occurs that prevents VTAM from reserving a session.

VTAM finds a session for the conversation as follows:

1. If a session is free, VTAM reserves it for a conversation.
2. If no free sessions exist and session limits allow, VTAM establishes a session and reserves it for a conversation.
3. If a new session cannot be established, VTAM queues the request until a session becomes available or until the session limits are changed to allow the establishment of a new session.

After session initiation, the conversation is reserved in PENDING ALLOCATE state and the application receives the conversation identifier in the CONVID field. The application could also receive the PCID for the session if VTRINA and VTRINL are specified on the preallocation request. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received on this conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is suspended for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
 [name] APPCCMD CONTROL=PREALLOC ,QUALIFY=ALLOCD (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
► [,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1)] ►
```

```
►──┬──────────────
```

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA

corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled

when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be preallocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NAMEUSE

Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME

The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME

The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*

Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD

Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN

Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY

Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM does

not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4-bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following values at the completion of this macroinstruction:
X'00' RESET
X'08' END_CONV
X'FF' PENDING_ALLOCATE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when

RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- Send FMH_5 sequence number vector (X'15')
- Receive FMH_5 sequence number vector (X'16')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

The conversation state is PENDING_ALLOCATE after successful completion of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_ NAME_REQUESTED
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	PARAMETER_ERROR—NETWORK- QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE.
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=PREALLOC, QUALIFY=CONVGRP

Purpose

This macroinstruction reserves a session for a conversation with a specified conversation group identifier without establishing a conversation. If the specified session is not available and session limits allow, VTAM queues the request until the session becomes available. If the specific session does not exist, VTAM fails the preallocation request. After a session is reserved, session related information can be passed between the application program and VTAM. The conversation is not active until the APPCCMD CONTROL=SENDFMH5 is issued.

Usage

QUALIFY=CONVGRP is used to preallocate a conversation over a specific session that already exists. It provides the ability to serially preallocate a related group of conversations on a particular session. This macroinstruction completes when:

- VTAM assigns the specified session to the conversation.
- The specific session is deactivated.
- An error occurs that prevents VTAM from assigning the session to the conversation.

To indicate the session to be used, the application program specifies the conversation group identifier for the session on the CGID keyword. The conversation group identifier of the session is returned to the application program by the CGID returned field for the following APPCCMD macroinstructions:

- APPCCMD CONTROL=ALLOC
- APPCCMD CONTROL=PREALLOC
- APPCCMD CONTROL=RCVFMH5

VTAM finds the session for the conversation as follows:

1. If the specified session is available, VTAM assigns it to the conversation.
2. If the specified session exists but is not available, VTAM queues the request until the session becomes available.
3. If the specified session is deactivated while the request is queued, the queued request is rejected.

After session initiation, the session is reserved to receive session related information if necessary and is assigned to a conversation. A conversation identifier is returned to the application in the CONVID field. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received on this conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
┌─────────┐
└name─────┘──APPCCMD CONTROL=PREALLOC──,QUALIFY=CONVGRP──────(2)──,RPL=──┬─rpl_address_field──────────(8)──┬──►
                                                                         └─(rpl_address_register)─────┘

►──┬──────────────
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CGID=*32-bit_conversation_group_id_field*

CGID=(*32-bit_conversation_group_id_register*)
Specifies the 32-bit conversation group ID.

This value can be obtained from a previous APPCCMD CONTROL=ALLOC, CONTROL=PREALLOC, or CONTROL=RCVFMH5 macroinstruction. If the CGID operand is not specified, VTAM uses the conversation group ID that is already in the RPL6CGID field on the RPL extension.

The conversation group ID identifies a specific session between two specific LUs. It provides a means by which a VTAM LU 6.2 application program and its partner LU can share serially the same session.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the subsequent APPCCMD CONTROL=SENDFMH5. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, for example, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, for example, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be

informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

NAMEUSE
Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME
The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME
The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4-bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were

supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPI extension.

This field can have the following values at the completion of this macroinstruction:
X'00' RESET
X'08' END_CONV
X'FF' PENDING_ALLOCATE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process

before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGMODE
Specifies the logon mode name designating the network properties for the session to be preallocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

LUNAME
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NETID
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located.

This network identifier is the identifier of the partner LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- Send FMH_5 sequence number vector (X'15')
- Receive FMH_5 sequence number vector (X'16')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')

- Partner's application capabilities vector (X'1A')

State Changes

The conversation state is PENDING_ALLOCATE after successful completion of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_ NAME_REQUESTED
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002A'	PARAMETER_ERROR—INVALID_CGID_VALUE_SPECIFIED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=PREALLOC, QUALIFY=CONWIN

Purpose

This macroinstruction reserves a contention-winner session for a conversation, if session limits allow, without establishing a conversation. If a contention-winner session is not available, VTAM queues the request for later completion. After a session is reserved, Session related information can be passed between the application program and VTAM. The conversation is not active until the APPCCMD CONTROL=SENDFMH5 is issued.

Usage

QUALIFY=CONWIN is used when an application program preallocates a conversation and wants VTAM to queue the request if no contention-winner session can be assigned. This macroinstruction completes when VTAM reserves a contention-winner session or an error occurs that prevents VTAM from assigning a session.

VTAM finds a session for the conversation as follows:

1. If a contention-winner session is currently available, VTAM reserves it for a conversation.
2. If no contention-winner session is available and session limits allow, VTAM establishes a new contention-winner session and assigns it to the conversation.
3. If a new contention-winner session cannot be established, VTAM queues the request until a contention-winner session is available or session limits are changed to allow a new contention-winner session to be activated.

For this macroinstruction to complete successfully, the session limits must define at least one contention-winner session.

If contention-winner sessions are deactivated under normal conditions and an APPCCMD CONTROL=PREALLOC, QUALIFY=CONWIN request is queued, VTAM activates another contention-winner session to meet the queued request.

After session initiation, the session is reserved to receive session related information if necessary and is assigned to a conversation. A conversation identifier is returned to the application in the CONVID field. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received on this conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►──┬──────────┬──APPCCMD CONTROL=PREALLOC──,QUALIFY=CONVGRP──────(2)──,RPL=──┬──rpl_address_field──────────┬──(8)──►
   └─name─────┘                                                              └─(rpl_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,AAREA=──┬──rpl_extension_address_field──────────┬──(1)──┘
              └─(rpl_extension_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,ACB=──┬──acb_address_field──────────┬──(1)──┘   └─,BRANCH=──┬──NO──┬──(1)──┘
            └─(acb_address_register)──────┘                       └──YES─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,CONMODE=──┬──BUFFCA──┬──(2)──┘   └─,CONXMOD=──┬──CA──┬──(2)──┘
                ├──CS──────┤                        └──CS──┘
                └──LLCA────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   ├─,ECB=──┬──INTERNAL──────────────────────────┬──(3)──(1)──┤
   │        ├──ecb_address_field──────────────────┤                                                              │
   │        └──(ecb_address_register)──────────────┘                                                              │
   │                                                                                                              │
   └─,EXIT=──┬──exit_routine_address_field──────────┬──(4)──(1)──┘
             └─(exit_routine_address_register)──────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,LOGMODE=8-byte_logon_mode_name──(2)──┘   └─,LUAFFIN=──┬──APPL──────┬──(2)──┘
                                              └──NOTAPPL───┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,LUNAME=8-byte_lu_name──(2)──┘   └─,NAMEUSE=──┬──APPNAME──┬──(2)──┘
                                                   └──GNAME────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,NETID=8-byte_network_identifier──(2)──┘   └─,OPTCD=(──┬──────────────────────────────────────────────────┬──)──┘
                                                             │                                                  │
                                                             │   ┌──────────────────────────────────────────┐   │
                                                             │   ↓                                          │   │
                                                             │   ┌──,──────────────────────────────────────┐  │   │
                                                             │   │                                         │  │   │
                                                             │   └──(7)──┬──ASY──(1)──────────────────────┬──┘   │
                                                             │           │                                 │      │
                                                             │           └──SYN──(1)───────────────────────┘      │
                                                             │                                                    │
                                                             │           ┌──KEEPSRB──(1)──(5)──────────────┐      │
                                                             │           │                                 │      │
                                                             │           └──NKEEPSRB──(1)──(6)─────────────┘      │
                                                             │                                                    │
                                                             └────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,RTSRTRN=──┬──BOTH──┬──(2)──┘   └─,USERFLD=──┬──4-bytes_of_user_data──┬──(2)──┘
                └──EXPD──┘                        └─(user_data_register)───┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                              │
   └─,VTRINA=──┬──vector_address_field──────────┬──(2)──┘
               └─(vector_address_register)──────┘
```

```
►──┬──────────────
```

under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE

Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA

Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS

Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA

Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD

Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB

Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL

Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be preallocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*

Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NAMEUSE

Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME

The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME

The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*

Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD

Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN

Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY

Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4-bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following values at the completion of this macroinstruction:
X'00' RESET
X'08' END_CONV
X'FF' PENDING_ALLOCATE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- Send FMH_5 sequence number vector (X'15')
- Receive FMH_5 sequence number vector (X'16')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')

- Partner's application capabilities vector (X'1A')

State Changes

The conversation state is PENDING_ALLOCATE after successful completion of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_ NAME_REQUESTED
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	PARAMETER_ERROR—NETWORK- QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH

RCPRI	RCSEC	Meaning
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=PREALLOC, QUALIFY=IMMED

Purpose

This macroinstruction reserves an active contention-winner session for a conversation, if session limits allow, without establishing a conversation. If no session is available, the preallocation request fails. After a session is reserved, session related information can be passed between the application program and VTAM. The conversation is not active until the APPCCMD CONTROL=SENDFMH5 is issued.

Usage

QUALIFY=IMMED is used to preallocate a conversation when the application program needs an immediate response from VTAM. This macroinstruction completes successfully only when an active contention-winner session is available to be reserved for a conversation. If the request cannot be met immediately, VTAM does not queue it. VTAM neither tries to activate a new session nor bids on a contention-loser session.

When a conversation is preallocated, VTAM assigns a conversation identifier to it. This identifier is returned in the CONVID field. The application program must associate a conversation with a particular transaction by using the conversation identifier.

The application program can specify how expedited data is to be received.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►
  [name] APPCCMD CONTROL=PREALLOC ,QUALIFY=IMMED (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►
  ,AARREA= rpl_extension_address_field (1)
           (rpl_extension_address_register)
►
  ,ACB= acb_address_field (1)
        (acb_address_register)
  ,BRANCH= NO (1)
           YES
►
  ,CONMODE= BUFFCA (2)
            CS
            LLCA
  ,CONXMOD= CA (2)
            CS
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                   (3)   (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────────┤
   │        └─(ecb_address_register)──────────┘
   │                                                                                                   (4)   (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)──────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                   (2)
   └─,LOGMODE=8-byte_logon_mode_name──────────────────────────────────────────────────────────────────────────────────────────┘
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                   (2)
   └─,LUNAME=8-byte_lu_name──────────────────────────────────────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                   (2)
   └─,NETID=8-byte_network_identifier─────────────────────────────────────────────────────────────────────────────────────────┘
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  ┌──────────────────────────────────────────────────────────────────┐
   │                                                  │                                   (7)                            │
   │                                                  ▼                                                                  │
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────)─┘
               │                                   (1)
               ├─┬─ASY──────────────────────────────┤
               │ │                                 (1)
               │ └─SYN──────────────────────────────┘
               │                                   (1)   (5)
               ├─┬─KEEPSRB────────────────────────────┤
               │ │                                 (1)   (6)
               │ └─NKEEPSRB─────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                   (2)
   └─,RTSRTRN=──┬─BOTH──────────────────────────────┤
                └─EXPD──────────────────────────────┘
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                   (2)
   └─,USERFLD=──┬─4-bytes_of_user_data──────────────┤
                └─(user_data_register)──────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                   (2)
   └─,VTRINA=──┬─vector_address_field──────────────┤
               └─(vector_address_register)──────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                                                                   (2)
   └─,VTRINL=──┬─vector_length_field───────────────┤
               └─(vector_length_register)───────────┘
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.
5. KEEPSRB is meaningful only for synchronous operations.
6. NKEEPSRB is meaningful only for synchronous operations.
7. You can code more than one suboperand on OPTCD, but no more than one from each group.
8. See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-

any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be preallocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA*

Programmer's LU 6.2 Guide for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NETID=*8-byte_network_identifier*
Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register*)
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4-bytes_of_user_data*

USERFLD=(*user_data_register*)
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register*)
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register*)
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following value at the completion of this macroinstruction:
X'00' RESET
X'08' END_CONV
X'FF' PENDING_ALLOCATE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension. The format of the session instance identifier is described in the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* .

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
Indicates that the session was established using session-level LU-LU verification.

NO (B'0')
Indicates that the session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- Send FMH_5 sequence number vector (X'15')
- Receive FMH_5 sequence number vector (X'16')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

The conversation state is PENDING_ALLOCATE after successful completion of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	PARAMETER_ERROR—NETWORK-QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0058'	X'0000'	UNSUCCESSFUL,_SESSION_NOT_AVAILABLE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED

RCPRI	RCSEC	Meaning
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE.
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=PREALLOC, QUALIFY=WHENFREE

Purpose

This macroinstruction reserves a session for a conversation, if session limits allow, without establishing a conversation. If a session is not available and one cannot be activated, VTAM returns control to the application program. After a session is reserved, session related information can be passed between the application program and VTAM. The conversation is not active until the APPCCMD CONTROL=SENDFMH5 is issued.

Usage

QUALIFY=WHENFREE is used when an application program preallocates a conversation and wants VTAM to search for a session that satisfies the ALLOCATE request. This macroinstruction completes when VTAM reserves a session for a conversation or when VTAM cannot reserve a session and returns control to the application program with a return code of X'0004', X'0001'.

VTAM finds a session for the conversation as follows:

1. If a session is available, VTAM reserves it for a conversation.
2. If no available sessions exist and session limits allow, VTAM establishes a session and reserves it for a conversation.
3. If a session cannot be established and session activation requests are pending, VTAM queues the PREALLOCATE request until the request is satisfied or until all pending session activation requests are used. If the pending session activation requests are used before the PREALLOCATE request is satisfied, VTAM fails the PREALLOCATE request with an RCPRI, RCSEC code of X'0004', X'0001'.
4. If a session cannot be established and no session activation request is pending that might satisfy the PREALLOCATE request, VTAM fails the PREALLOCATE request with an RCPRI, RCSEC code of X'0004', X'0001' and returns control to the application program.

After session initiation, the session is reserved to receive session related information if necessary and is assigned to a conversation. When a conversation is preallocated, VTAM assigns a conversation identifier to it. This identifier is returned in the CONVID field. The application program associates a conversation with a particular transaction by using the conversation identifier (CONVID).

The application program can specify how expedited data is to be received on this conversation.

Context

This macroinstruction is independent of conversation states when it is issued. The initial conversation state is created after this macroinstruction completes.

When a mode is retained for persistent LU-LU sessions, this macroinstruction is not allowed.

Syntax

```
►►                                                  (2)                          (8)
┬──────┬──APPCCMD CONTROL=PREALLOC──,QUALIFY=WHENFREE──────,RPL=──┬─rpl_address_field──────┬──►
└name  ┘                                                         └(rpl_address_register)──┘
```

```
►──┬──────────────
```

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY

can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled

when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOGMODE=*8-byte_logon_mode_name*
Specifies the logon mode name designating the network properties for the session to be preallocated for this conversation. The network properties include, for example, the class of service to be used.

The logon mode name cannot be blanks. The logon mode name can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads it on the right with blanks.

If the logon mode parameter on the APPCCMD macroinstruction specifies a logon mode name that does not exist in the logon mode table, VTAM uses the mode name of blanks to retrieve the default mode entry when processing session activation requests. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.) This logon mode name corresponds to a logon mode name specified in a MODEENT definition statement. (The MODEENT statement is used to build the logon mode table named in the MODETAB operand of the APPL definition statement for this application program.) For more information on the MODEENT macroinstruction, refer to *z/OS Communications Server: SNA Resource Definition Reference*. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
Specifies the name of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is located. This LU name is the network name of the target LU. It can be up to eight characters in length. If it is less than eight characters in length, VTAM pads the LU name on the right with blanks. It is labeled RPL6LU in the RPL extension.

NAMEUSE

Specifies the preferred type of name identifying the application to the partner LU in the PLU name structured user data subfield in the BIND requests or in the SLU name structured user data subfield in BIND responses sent while the application is acting as a generic resource.

NAMEUSE=APNAME

The application identifies itself to the partner LU by its application network name.

NAMEUSE=GNAME

The application identifies itself to the partner LU by a generic resource name.

The NAMEUSE value is honored if no sessions currently exist with the partner LU and if no partner affinity is being retained. If any active or pending sessions exist or a partner affinity is being retained, the previous type of name is used for new sessions. If NAMEUSE is not specified, the generic resource name will be the preferred name used when starting sessions as a generic resource.

NETID=*8-byte_network_identifier*

Specifies the network identifier of the partner application program at which the remote transaction program, specified in the FMH-5 supplied in the AREA field, is found.

If PARMS= (NQNAMES=NO) is specified on the ACB macroinstruction and you specify NETID, the NETID value is ignored. If PARMS= (NQNAMES=YES) is specified on the ACB macroinstruction, NETID must be supplied.

If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified name of the target LU. (If NETID is specified, LUNAME must be specified.)

This network identifier is the identifier of the target LU. It can be up to eight characters in length. If it is fewer than eight characters in length, VTAM pads the network identifier on the right with blanks. It is labeled RPL6NET in the RPL extension.

OPTCD

Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN

Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY

Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM does

not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4-bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

AVFA
The field in the RPL extension that indicates whether the partner LU accepts the already-verified indicator in place of the password security access subfield on the FMH-5s that it receives. This field is labeled RPL6AVFA in the RPL extension.

YES (B'1')
The partner LU accepts the already-verified indicator.

NO (B'0')
The partner LU does not accept the already-verified indicator.

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following value at the completion of this macroinstruction:
X'00' RESET
X'08' END_CONV
X'FF' PENDING_ALLOCATE

CONVID
Specifies the resource identifier of the conversation. This field is labeled RPL6CNVD in the RPL extension.

Note: The value in this field is returned before this macroinstruction completes to allow the application to cancel the conversation allocation process before it completes. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

CONVSECP
The field in the RPL extension that returns an indication of whether the partner LU accepts FMH-5s that include security subfields and indicators. The indication is either YES or NO (RPL6CLSA in RPL6RTUN set on or off). This field is labeled RPL6CLSA in the RPL extension.

YES (B'1')
The partner LU accepts FMH-5s with security subfields and indicators. The subfields allow the application program to include a password, user ID, and profile on allocation requests.

NO (B'0')
The partner LU does not accept FMH-5s with security subfields. If this is the case, VTAM strips out any security subfields and indicators that might be included on an allocation request.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LUAFFIN
The field in the RPL extension that indicates the requested (on input) or actual (on output) ownership of a Generic Resource affinity with the partner LU, if one exists. A result value is only returned at completion if a requested value is specified when the macroinstruction is issued.

NONE (B'00')
GR affinity is not applicable or is unknown.

NOTAPPL (B'01')
GR affinity is not application-owned.

APPL (B'10')
GR affinity is application-owned.

PRSISTVP
Indicates that the partner LU accepts requests for persistent verification. This field is labeled RPL6PV in the RPL extension.

YES (B'1')
The partner LU accepts persistent-verification indicators.

NO (B'0')
The partner LU does not accept persistent-verification indicators.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when

RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. However, not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. If the APPCCMD failed because an attempt to establish a session failed, this field contains a sense code indicating the cause of the failure. This field is labeled RPL6SNSI in the RPL extension.

SESSID
The field in the RPL extension that returns a session instance identifier of the session over which the FMH-5 flows. The FMH-5 is supplied by the application program using the AREA field. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range of 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- Send FMH_5 sequence number vector (X'15')
- Receive FMH_5 sequence number vector (X'16')
- PCID vector (X'17')
- Name change vector (X'18')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

The conversation state is PENDING_ALLOCATE after successful completion of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'000A'	SESSIONS_WILL_USE_APPL_NAME_GENERIC_ NAME_REQUESTED
X'0000'	X'000B'	SESSIONS_WILL_USE_GENERIC_NAME_APPL_ NAME_REQUESTED
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0001'	ALLOCATION_ERROR—ALLOCATION_FAILURE_RETRY
X'0004'	X'000E'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'0004'	X'000F'	DEALLOCATION_REQUESTED
X'002C'	X'0000'	PARAMETER_ERROR—INVALID_LU_NAME_OR_ NETWORK_IDENTIFIER
X'002C'	X'0001'	PARAMETER_ERROR—INVALID_MODE
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002B'	PARAMETER_ERROR—NETWORK- QUALIFIED_NAME_REQUIRED
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE.
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B0'	X'0001'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ VARIANT_NAME_ENTRY
X'00B0'	X'0002'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ DIFFERS_FROM_ASSOCIATED_NAME
X'00B0'	X'0003'	NAME_RESOLUTION_ERROR—NAME_RETURNED_FOUND_ IN_VARIANT_NAME_ENTRY
X'00B0'	X'0004'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_SUPPLIED_NAME_ENTRY
X'00B0'	X'0005'	NAME_RESOLUTION_ERROR—PARTNER_NETWORK_ NAME_MISMATCH
X'00B0'	X'0006'	NAME_RESOLUTION_ERROR—LUNAME_FOUND_IN_ UNUSABLE_NAME_ENTRY
X'00B0'	X'0007'	NAME_RESOLUTION_ERROR—NAME_RETURNED_ FOUND_IN_UNUSABLE_NAME_ENTRY
X'00B0'	X'0008'	NAME_RESOLUTION_ERROR—LU_NAME_FOUND_ IN_A_DISASSOCIATED_NAME_ENTRY

APPCCMD CONTROL=PREPRCV, QUALIFY=CONFIRM

Purpose

This macroinstruction is used to change the local conversation state of a half-duplex conversation from SEND to RECEIVE. This macroinstruction flushes the SEND buffer and then sends a confirmation request to the partner application program. When a positive acknowledgment to the confirmation is received, the macroinstruction changes the conversation state from SEND to RECEIVE.

Usage

This macroinstruction synchronizes the communication between the local and remote LUs. It is issued when the application program has finished sending and is ready to receive. This macroinstruction causes VTAM to flush the SEND buffer (in the same way as it does for APPCCMD CONTROL=SEND, QUALIFY=CONFIRM) and send a confirmation request to the partner LU.

If a positive acknowledgment to the confirmation is received (as indicated by an RCPRI of X'0000'), VTAM changes the conversation from SEND to RECEIVE state in preparation to receive data. If a negative confirmation response is received (RCPRI is not X'0000'), the state of the conversation is found in the CONSTATE field.

This macroinstruction corresponds to the PREPARE_TO_RECEIVE (TYPE=CONFIRM) verb described in the LU 6.2 architecture.

Context

This macroinstruction can be issued only from SEND or PENDING_SEND conversation state. This macroinstruction is not allowed on full-duplex conversations.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

The local application can specify whether this acknowledgment is a response (LOCKS=SHORT) or data received from the partner (LOCKS=LONG). The LOCKS=SHORT specification completes more quickly and the LOCKS=LONG specification uses fewer transmission flows and processing cycles.

Syntax

```
►►
 [name] APPCCMD CONTROL=PREPRCV ,QUALIFY=CONFIRM (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
► [ ,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1) ] ►
► [ ,ACB= [acb_address_field | (acb_address_register)] (1) ] [ ,BRANCH= [NO | YES] (1) ] ►
```

```
►──┬──────────────
```

AAREA=(*rpl_extension_address_register*)
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register*)
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record

format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOCKS
Specifies when the execution of the macroinstruction is complete following execution of the CONFIRM function. This field corresponds to the LOCKS

parameter on the PREPARE_TO_RECEIVE verb, as described in the LU 6.2 architecture. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information on the use of this function.) This field is labeled RPL6LOCK in the RPL extension.

LOCKS=LONG
Specifies that the function of this macroinstruction is complete when information, such as data, is received from the partner application. The receipt of data presumes an affirmative reply to the confirmation request. The local application program must issue an APPCCMD CONTROL=RECEIVE in order to get the information that caused the macroinstruction to complete.

LOCKS=SHORT
Specifies that the function of this macroinstruction is complete when a positive response is received to the confirmation request.

Note: The partner cannot determine whether LOCKS=LONG or SHORT was specified. The APPCCMD CONTROL=SEND, QUALIFY=CONFRMD must be specified in either case.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0

- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

This field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. It is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log

data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner application program. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not recognized by VTAM. This field is labeled RPL6SNSI in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

The conversation state is RECEIVE after successful processing.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD

RCPRI	RCSEC	Meaning
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ NOT_VALID_FOR_FULL-DUPLEX_CONVERSATIONS
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=PREPRCV, QUALIFY=DATACON

Purpose

This macroinstruction sends data, flushes the SEND buffer, and then sends a confirmation request to the partner application program. If a positive confirmation acknowledgment is received, the local conversation state is changed from SEND to RECEIVE state.

Usage

This macroinstruction combines the functions of two macroinstructions, APPCCMD CONTROL=SEND, QUALIFY=DATA followed by APPCCMD CONTROL=PREPRCV, QUALIFY=CONFIRM. VTAM flushes the SEND buffer and sends the data that is specified on the macroinstruction. A confirmation request follows. The application program must ensure that the data sent completes a logical record.

If a positive acknowledgment to the confirmation request is received, the conversation is placed in RECEIVE state. When this macroinstruction completes without error, the state of the conversation is contained in the CONSTATE field.

This macroinstruction corresponds to the verbs SEND_DATA followed by PREPARE_TO_RECEIVE (TYPE=CONFIRM) described in the LU 6.2 architecture.

Context

This macroinstruction can be issued from the SEND or PENDING_SEND conversation state. This macroinstruction is not allowed on full-duplex conversations.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=PREPRCV ,QUALIFY=DATACON (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►
 ,AAREA= rpl_extension_address_field (1)
         (rpl_extension_address_register)
►
 ,ACB= acb_address_field (1)
       (acb_address_register)
►
 ,AREA= data_area_or_buffer_list_address_field (1)
        (data_area_or_buffer_list_address_register)
 ,BRANCH= NO (1)
          YES
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                                 (2)
   └─,CONMODE=──┬─BUFFCA──┬──┘   └─,CONVID=──┬─32-bit_resource_id_field──┬──┘
                ├─CS──────┤                  └─(32-bit_resource_id_register)─┘
                ├─LLCA────┤
                └─SAME────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                                 (1)
   └─,CONXMOD=──┬─CA──────┬──┘   └─,CRYPT=──┬─NO──┬──┘
                ├─CS──────┤                  └─YES─┘
                └─SAME────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                    (3)    (1)                                                                      (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────────────────────────────────────────────────────────────────────────────────────────────────
   │        └─(ecb_address_register)──────────────────────────────────────────────────────────────────────────────────────────────────
   │                                                                                                                                   (4)    (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)──────────────────────────────────────────────────────────────────────────────────────────

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (7)                                                                                                 (1)
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)
               ├─ASY──────────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)
               │                                  (1)
               ├─SYN──────────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)
               ├─BUFFLST──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)
               ├─NBUFFLST─────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)
               ├─XBUFLST──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)    (5)
               ├─KEEPSRB──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                  (1)    (6)
               └─NKEEPSRB─────────────────────────────────────────────────────────────────────────────────────────────────────────────
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.
5. KEEPSRB is meaningful only for synchronous operations.
6. NKEEPSRB is meaningful only for synchronous operations.
7. You can code more than one suboperand on OPTCD, but no more than one from each group.
8. See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

  If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

LOCKS
Specifies when the execution of the macroinstruction is complete following execution of the CONFIRM function. This field corresponds to the LOCKS parameter on the PREPARE_TO_RECEIVE verb, as described in the LU 6.2 architecture. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information on the use of this function.) This field is labeled RPL6LOCK in the RPL extension.

LOCKS=LONG
Specifies that the function of this macroinstruction is complete when information, such as data, is received from the partner application. The receipt of data presumes an affirmative reply to the confirmation request. The local application program must issue an APPCCMD CONTROL=RECEIVE in order to get the information that caused the macroinstruction to complete.

LOCKS=SHORT
Specifies that the function of this macroinstruction is complete when a positive response is received to the confirmation request.

Note: The partner cannot determine whether LOCKS=LONG or SHORT was specified. The APPCCMD CONTROL=SEND, QUALIFY=CONFRMD must be specified in either case.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained

within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.
- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. It is labeled RPL6CCST in the RPL extension.

This field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG

X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. It is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue an APPCCMD CONTROL=RCVFMH5 to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner application program. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:
X'0004'
ALLOCATION_ERROR
X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE
X'001C'
DEALLOCATE_ABEND_TIMER
X'0030'
PROGRAM_ERROR_NO_TRUNC
X'0034'
PROGRAM_ERROR_PURGING
X'0038'
PROGRAM_ERROR_TRUNC
X'003C'
SERVICE_ERROR_NO_TRUNC
X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field

indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not recognized by VTAM. This field is labeled RPL6SNSI in the RPL extension.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. It is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). This field is labeled RPL6RSIG in the RPL extension.

YES (B'1')
A SIGNAL RU has been received from the partner application program. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner application program. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the

conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

The conversation enters RECEIVE state after successful completion of the macroinstruction.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (REMOTE PROGRAM REPLIED AFFIRMATIVELY)
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT

RCPRI	RCSEC	Meaning
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ NOT_VALID_FOR_FULL-DUPLEX_CONVERSATIONS
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=PREPRCV, QUALIFY=DATAFLU

Purpose

This macroinstruction sends data to a partner LU and flushes the SEND buffer. The conversation state for the application program is then changed from SEND to RECEIVE.

Usage

This macroinstruction combines the functions of two macroinstructions, APPCCMD CONTROL=SEND, QUALIFY=DATA followed by APPCCMD CONTROL=PREPRCV, QUALIFY=FLUSH. VTAM sends any data currently in the SEND buffer. This data is followed by the data specified on the macroinstruction to the partner LU. The application program must ensure that the data sent completes a logical record.

If the data is sent successfully, the conversation is placed in RECEIVE state. The conversation state is found in the CONSTATE field when the macroinstruction completes.

This macroinstruction corresponds to the SEND_DATA followed by PREPARE_TO_RECEIVE (TYPE=FLUSH) verbs described in the LU 6.2 architecture.

Context

This macroinstruction can be issued from the SEND or PENDING_SEND conversation state. This macroinstruction is not allowed on full-duplex conversations.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=PREPRCV ,QUALIFY=DATAFLU (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►
 ,AAREA= rpl_extension_address_field (1)
         (rpl_extension_address_register)
►
 ,ACB= acb_address_field (1)
       (acb_address_register)
►
 ,AREA= data_area_or_buffer_list_address_field (1)
        (data_area_or_buffer_list_address_register)
 ,BRANCH= NO (1)
          YES
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                          │
   └─,CONMODE=──┬─BUFFCA──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────
                │─CS──────│                                                                                                          │
                │─LLCA────│                                                                                                          │
                └─SAME────┘                                                                                                          │
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                  │
   │                                                                                                                                  │
   │                                                                                                                                  │
```

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

  If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.
- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension. It can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE-ONLY_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5

has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that VTAM did not recognize.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. It is labeled RPL6RSIG in the RPL extension. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list entry (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list entry (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

The conversation state is RECEIVE after successful processing.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH

RCPRI	RCSEC	Meaning
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ ALLOWED_ON_MODE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0004'	CONTROL/QUALIFY_VALUE_NOT_VALID_FOR_ FULL-DUPLEX_CONVERSATIONS

RCPRI	RCSEC	Meaning
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=PREPRCV, QUALIFY=FLUSH

Purpose

This macroinstruction flushes the application program's SEND buffer and changes the conversation state from SEND to RECEIVE.

Usage

This macroinstruction executes the function of the APPCCMD CONTROL=SEND, QUALIFY=FLUSH macroinstruction. The application program must ensure that the data in the SEND buffer completes a logical record.

If the data is sent successfully, the conversation is put in RECEIVE state. The conversation state is in the CONSTATE field when the macroinstruction completes.

This macroinstruction corresponds to the PREPARE_TO_RECEIVE (TYPE=FLUSH) verb described in the LU 6.2 architecture.

Context

This macroinstruction can be issued from the SEND or PENDING_SEND conversation state. This macroinstruction is not allowed on full-duplex conversations.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
  [name] APPCCMD CONTROL=PREPRCV ,QUALIFY=FLUSH (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_field (1)
        (acb_address_register)
  ,BRANCH= NO (1)
           YES
►
  ,CONMODE= BUFFCA (2)
            CS
            LLCA
            SAME
  ,CONVID= 32-bit_resource_id_field (2)
           (32-bit_resource_id_register)
►
  ,CONXMOD= CA (2)
            CS
            SAME
  ,ECB= INTERNAL (3) (1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field (4) (1)
         (exit_routine_address_register)
```

```
►──┬──────────────
```

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension. It can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

The conversation state is RECEIVE after successful processing.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. Refer to “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ NOT_VALID_FOR_FULL-DUPLEX_CONVERSATIONS
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY

Purpose

This macroinstruction receives expedited information from any active conversation whose expedited information mode is continue-any. VTAM will wait for expedited information to arrive on a conversation in continue-any mode to satisfy the macroinstruction request.

Usage

This macroinstruction can be used when the application program is maintaining multiple asynchronous conversations. Instead of issuing APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC for each conversation, the application program can put the conversations in continue-any mode for receiving expedited information and issue a single APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY.

The application program must receive the entire amount of expedited data available. If the length of the area specified by the application is not sufficient to receive all the expedited data available, an RCPRI,RCSEC combination of PARAMETER_ERROR_SUPPLIED_ LENGTH_INSUFFICIENT is returned to the application. The maximum amount of data that can be received from the partner is 86 bytes.

A Request_To_Send_Received indication is sufficient to complete this macroinstruction. If a Request_To_Send_Received indication and expedited data are present, then both will be returned to the application. The settings of the SIGRCV and SIGDATA returned parameter fields will indicate whether a Request_To_Send_Received indication (Signal Data) was received on the conversation. When expedited data is available on a conversation whose expedited information mode is continue-any, VTAM copies the data into the data area that is specified on the AREA parameter and completes the macroinstruction. The conversation identifier of the conversation that satisfied the macroinstruction is placed in the CONVID field.

Multiple APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY macroinstructions can be outstanding concurrently. The macroinstruction can be issued when no conversations exist that are in continue-any mode for receiving expedited information. VTAM queues the APPCCMD until one or more conversations are placed in continue-any mode for receiving information and has expedited information available to be received.

Context

Input states are not applicable to this macroinstruction. Only expedited information for a conversation that is not in PENDING_DEALLOCATE, END_CONV or FDX_RESET state and whose expedited information mode is continue-any satisfies this type of RCVEXPD.

Syntax

```
►►                                                  (2)                                  (8)
┌──────────┐──APPCCMD CONTROL=RCVEXPD──,QUALIFY=ANY──────,RPL=┬─rpl_address_field──────────┬──►
└─name─────┘                                                 └─(rpl_address_register)─────┘
```

```
►──┬──────────────
```

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register*)
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension. For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG

X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

CONVID
Specifies the resource identifier of the conversation on which information was received. A value is placed in this field by VTAM only if QUALIFY=ANY. This field is labeled RPL6CNVD in the RPL extension.

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of expedited data the application program received. If the application program receives information other than data, this variable is set to 0. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture. The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_ INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD

APPCCMD CONTROL=RCVEXPD, QUALIFY=IANY

Purpose

This macroinstruction receives expedited information from any active conversation whose expedited information mode is continue-any. VTAM will not wait for expedited information to arrive on a conversation in continue-any mode to satisfy the macroinstruction.

Usage

This macroinstruction can be used when the application program is maintaining multiple asynchronous conversations. Instead of issuing APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC for each conversation, the application program can put the conversations in continue-any mode for receiving expedited information and issue a single APPCCMD CONTROL=RCVEXPD, QUALIFY=IANY.

A Request_To_Send_Received indication is sufficient to successfully complete this macroinstruction. If a Request_To_Send_Received indication and expedited data are present then both will be returned to the application. The settings of the SIGRCV and SIGDATA returned parameter fields will indicate whether a Request_To_Send_Received indication (Signal Data) was received on the conversation. When expedited data is available on a conversation whose expedited information mode is continue-any, VTAM copies the data into the data area that is specified on the AREA parameter and completes the macroinstruction. The conversation identifier of the conversation that satisfied the macroinstruction is placed in the CONVID field.

When issued and no conversation exists in a continue-any mode for expedited data or no conversations in continue-any mode have received expedited information, an RCPRI, RCSEC combination of X'0000', X'0008', NO_IMMEDIATELY_AVAILABLE_INFORMATION is returned to the application program.

The application must receive the entire amount of expedited data available. If the length of the area specified by the application is not sufficient to receive all the expedited data available, an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT is returned to the application.

Context

Input states are not applicable to this macroinstruction. Only expedited information for a conversation that is not in PENDING_DEALLOCATE, END_CONV, or FDX_RESET and whose expedited information mode is continue-any satisfies this type of RCVEXPD.

Syntax

```
►►                                                  (2)                          (8)
┬──────────┬APPCCMD CONTROL=RCVEXPD──,QUALIFY=IANY────────,RPL=─┬─rpl_address_field──────────┬──►
└─name─────┘                                                   └─(rpl_address_register)─────┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                    │
   └─,AAREA=──┬─rpl_extension_address_field──────────────────────────────────────────────────────────────────────────────
```

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register*)
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG

X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

CONVID
Specifies the resource identifier of the conversation on which information was received. A value is placed in this field by VTAM only if QUALIFY=ANY|IANY. This field is labeled RPL6CNVD in the RPL extension.

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of expedited data the application program received up to the maximum. If the application program receives information other than data, this variable is set to 0. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture. The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0008'	NO_IMMEDIATELY_AVAILABLE_INFORMATION
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_ INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RCVEXPD, QUALIFY=ISPEC

Purpose

This macroinstruction receives expedited information immediately available on a specified conversation. VTAM will not wait for expedited information to arrive to satisfy the macroinstruction request.

Usage

A Request_To_Send_Received indication is sufficient to successfully complete this macroinstruction. The conversation mode (CONXMOD) for expedited data may be either CA or CS. If a Request_To_Send_Received indication and expedited data are present then both will be returned to the application. The settings of the SIGRCV and SIGDATA returned parameter fields will indicate whether a Request_To_Send_Received indication (Signal Data) was received on the conversation. When expedited data is available on the conversation, VTAM copies the data into the data area that is specified on the AREA parameter and completes the macroinstruction.

If expedited information is not available, an RCPRI, RCSEC combination of X'0000', X'0008', NO_IMMEDIATELY_AVAILABLE_INFORMATION is returned to the application.

The application must receive the entire amount of expedited data available. If the length of the area specified by the application is not sufficient to receive all the expedited data available, an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT is returned to the application.

If this macroinstruction is issued while another RCVEXPD macroinstruction is currently outstanding for the specified conversation, an RCPRI, RCSEC combination of X'002C', X'0011', PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING is returned to the application. The maximum amount of expedited data that can be received is 86 bytes.

If the RECEIVE EXPEDITED queue has been prohibited, then an RCPRI, RCSEC combination of X'00A0', X'0002', REQUEST_NOT_ALLOWED—REQUEST_BLOCKED is returned to the application. The RECEIVE EXPEDITED queue is prohibited when the conversation is in the process of being deallocated or terminated.

If the macroinstruction is issued for a conversation in PENDING_DEALLOCATE state, an RCPRI, RCSEC combination of X'0050', X'0000', STATE_ERROR is returned to the conversation.

If the conversation ends before this macroinstruction can process, an RCPRI, RCSEC combination of X'0000', X'0009', REQUEST_TERMINATED_BY_END_OF_CONVERSATION is returned to the application.

This macroinstruction corresponds to the RECEIVE_EXPEDITED_DATA (IMMEDIATE) verb described in the LU 6.2 architecture.

Context

This macroinstruction can be issued from any conversation state except PENDING_DEALLOCATE, END_CONV, or FDX_RESET.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
  [name] APPCCMD CONTROL=RCVEXPD ,QUALIFY=ISPEC (2) ,RPL= rpl_address_field (8)
                                                          (rpl_address_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,AAREA= rpl_extension_address_field (1)
  │        (rpl_extension_address_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,ACB= acb_address_field (1)
  │      (acb_address_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,AREA= data_area_address_field (1)
  │       (data_area_address_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,AREALEN= data_area_length (1)
  │          (data_area_length_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,BRANCH= NO (1)
  │         YES
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,CONMODE= BUFFCA (2)
  │          CS
  │          LLCA
  │          SAME
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,CONVID= 32-bit_resource_id_field (2)
  │         (32-bit_resource_id_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,CONXMOD= CA (2)
  │          CS
  │          SAME
►─┬──────────────────────────────────────────────────────────────────────────────────►
  │,ECB= INTERNAL (3) (1)
  │      ecb_address_field
  │      (ecb_address_register)
  │,EXIT= exit_routine_address_field (4) (1)
  │       (exit_routine_address_register)
►─┬──────────────────────────────────────────────────────────────────────────────────►◄
  │,OPTCD=( ,(7) )
  │         ASY (1)
  │         SYN (1)
  │         KEEPSRB (1) (5)
  │         NKEEPSRB (1) (6)
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of expedited data the application program received. The value returned will always be less than or equal to the value specified for AREALEN. This value will be set to 0 if the macroinstruction is being completed because of a REQUEST_TO_SEND being received.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture. The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

There are no state changes caused by the execution of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0008'	NO_IMMEDIATELY_AVAILABLE_INFORMATION
X'0000'	X'0009'	REQUEST_TERMINATED_BY_END_OF_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_ INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC

Purpose

This macroinstruction receives expedited information from the specified conversation. VTAM will wait for expedited information to arrive to satisfy the macroinstruction request. If expedited information is immediately available, then the application receives it without waiting. The expedited information mode may be continue-any or continue-specific.

Usage

A Request_To_Send_Received indication is sufficient to successfully complete this macroinstruction. If a Request_To_Send_Received indication and expedited data are present then both will be returned to the application. The settings of the SIGRCV and SIGDATA returned parameter fields will indicate whether a Request_To_Send_Received indication (Signal Data) was received on the conversation. When expedited data is available on the conversation, VTAM copies the data into the data area that is specified on the AREA parameter and completes the macroinstruction.

The application must receive the entire amount of expedited data available. If the length of the area specified by the application program is not sufficient to receive all the expedited data available, an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR_SUPPLIED_LENGTH_INSUFFICIENT is returned to the application. The maximum amount of expedited data that can be received is 86 bytes.

If this macroinstruction is issued while another RCVEXPD macroinstruction is currently outstanding for the specified conversation, an RCPRI, RCSEC combination of X'002C', X'0011', PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING is returned to the application.

If the RECEIVE EXPEDITED queue has been prohibited, then an RCPRI, RCSEC combination of X'00A0', X'0002', REQUEST_NOT_ALLOWED—REQUEST_BLOCKED is returned to the application. The RECEIVE EXPEDITED queue is prohibited when the conversation is in the process of being allocated or terminated.

If the macroinstruction is issued for a half-duplex conversation and a negative response is received from the partner, then an RCPRI, RCSEC combination of X'00A0', X'0003', REQUEST_NOT_ALLOWED—EXECUTION_OF_REQUEST_TERMINATED will be returned to the application.

If the macroinstruction is issued for a conversation in PENDING_DEALLOCATE state, an RCPRI, RCSEC combination of X'0050', X'0000', STATE_ERROR is returned to the application.

If the conversation is terminated before expedited information is received, an RCPRI, RCSEC combination of X'0000', X'0009', REQUEST_TERMINATED_BY_END_OF_CONVERSATION is returned to the application.

This macroinstruction corresponds to the RECEIVE_EXPEDITED_DATA (WHEN_EXPEDITED_DATA_RECEIVED) verb described in the LU 6.2 architecture.

Context

This macroinstruction can be issued from any conversation state except PENDING_DEALLOCATE, END_CONV, or FDX_RESET.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=RCVEXPD ,QUALIFY=SPEC(2) ,RPL= rpl_address_field(8)
                                                       (rpl_address_register)
►
 ,AARREA= rpl_extension_address_field(1)
          (rpl_extension_address_register)
►
 ,ACB= acb_address_field(1)
       (acb_address_register)
►
 ,AREA= data_area_address_field(1)
        (data_area_address_register)
►
 ,AREALEN= data_area_length(1)                ,BRANCH= NO(1)
           (data_area_length_register)                YES
►
 ,CONMODE= BUFFCA(2)                          ,CONVID= 32-bit_resource_id_field(2)
           CS                                          (32-bit_resource_id_register)
           LLCA
           SAME
►
 ,CONXMOD= CA(2)                              ,ECB= INTERNAL(3) (1)
           CS                                        ecb_address_field
           SAME                                      (ecb_address_register)
                                              ,EXIT= exit_routine_address_field(4) (1)
                                                     (exit_routine_address_register)
►
 ,OPTCD=( ,(7) )
          ASY(1)
          SYN(1)
          KEEPSRB(1) (5)
          NKEEPSRB(1) (6)
►◄
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a

state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register*)
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application

programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of expedited data the application program received. The value returned will always be less than or equal to the value specified for AREALEN. This value is set to 0 if the macroinstruction completes only due to receipt of a REQUEST_TO_SEND indication.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture. The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were

supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

There are no state changes caused by the execution of this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0009'	REQUEST_TERMINATED_BY_END_OF_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_ INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0003'	REQUEST_NOT_ALLOWED—EXECUTION_OF_ REQUEST_TERMINATED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RCVFMH5, QUALIFY=DATAQUE

Purpose

This macroinstruction receives an FMH-5, which begins the application program's participation in a conversation.

This macroinstruction allows the application to specify how expedited information is received.

Usage

When this macroinstruction is issued, VTAM copies the FMH-5, which represents a new conversation, into the area specified on the AREA parameter. When the macroinstruction completes, the new conversation identifier can be found in the CONVID field. The new conversation will be in RECEIVE state for half-duplex conversations and in SEND/RECEIVE state for full-duplex conversations.

If this macroinstruction is issued before an FMH-5 is received, VTAM waits until the FMH-5 is received to complete the macroinstruction. When an FMH-5 is received, VTAM bypasses the ATTN exit. If VTAM receives the FMH-5 before this macroinstruction is issued, VTAM schedules the ATTN exit. In either case, VTAM then moves the FMH-5 to the application's buffer and returns the CONVID and other return parameters.

After performing the operation of the RCVFMH5, VTAM examines the setting of the FILL parameter. If FILL=LL has been specified, this macroinstruction performs the functions of an APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC with a FILL=LL. That is, VTAM receives a single logical record. This would be the first logical record after the FMH-5. However, if FILL=BUFF has been specified, this macroinstruction performs the functions of an APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC with a FILL=BUFF. If there is insufficient information to complete the receive, the macroinstruction is suspended until more information is received from the partner. As is normally done for a receive macroinstruction, if any of the following occurs, the APPCCMD CONTROL=RCVFMH5, QUALIFY=DATAQUE completes:

- The local receive buffer is completely filled.
- A complete logical record is received AND FILL=LL was specified.
- A SEND indication is received.
- A CONFIRM indication is received.
- A DEALLOCATE indication is received.
- A ERROR condition is detected.

The application program can use the FMH-5 to perform conversation level security processing. Also, the FMH-5 indicates whether any GDS fields, such as DCE security or program initialization (PIP) data, follows the FMH-5.

Context

This macroinstruction can be issued from the RESET conversation state.

This macroinstruction is not mode-specific and might be issued for a mode that is retained for persistent LU-LU sessions. However, an FMH-5 is not returned for a mode that is being retained for persistent LU-LU sessions when this macroinstruction is issued.

Syntax

```
►►
 [name] APPCCMD CONTROL=RCVFMH5,QUALIFY=DATAQUE,RPL= rpl_address_field (8)
                                                     [(rpl_address_register)]
►
 [,AAREA= rpl_extension_address_field (1)
          [(rpl_extension_address_register)]]
►
 [,ACB= acb_address_field (1)
        [(acb_address_register)]]
►
 [,AREA= data_area_address_field (1)
         [(data_area_address_register)]]
►
 [,AREALEN= data_area_length (1)
            [(data_area_length_register)]]  [,BRANCH= NO (1)
                                                      [YES]]
►
 [,CD= DEFER (2)
       [IMMED]]  [,CONMODE= BUFFCA (2)
                            [CS]
                            [LLCA]]
►
 [,CONXMOD= CA (2)
            [CS]]
►
 [,ECB= INTERNAL (3) (1)
        [ecb_address_field]
        [(ecb_address_register)]
 [,EXIT= exit_routine_address_field (4) (1)
         [(exit_routine_address_register)]]
►
 [,FILL= BUFF (2)
         [LL]]  [,OPTCD=( ,
                          (7)
                          [ASY (1)
                          [SYN (1)]
                          [KEEPSRB (1) (5)
                          [NKEEPSRB (1) (6)])]
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                
   └─,RTSRTRN=──┬─BOTH─┬──────────────────────────────────────────────────────────────────────────────────────────────
                │      │                                                                                                
                └─EXPD─┘                                                                                                
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────────────────
   │                                                                                                                    
   └─,USERFLD=──┬─4-bytes_of_user_data─┬──────────────────────────────────────────────────────────────────────────────
                │                      │                                                                                
                └─(user_data_register)─┘                                                                                

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                    
   │                                                                                                                    
   │                                                                                                                    
```

AREA=(*data_area_address_register*)
Specifies the data area in which the application program is to receive the FMH-5 and any associated data. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register*)
Specifies the size of the supplied buffer area. An FMH-5 is, at most, 255 bytes in length. Since the application cannot determine the length of the FMH-5 when the RCVFMH5 request is queued, VTAM fails this macroinstruction if the length of AREALEN is less than 255 with an RCPRI, RCSEC combination of X'002C', X'0008'. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CD
Specifies whether the LU immediately goes to SEND or whether the LU defers the SEND transition by going into PEND_SEND when a change of direction is received with no data. This parameter is valid only for half-duplex conversations.

CD=DEFER
Specifies that the conversation state will be in PEND_SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CD=IMMED
Specifies that the conversation state will be in SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It

indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=*user_data_register*
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0

- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONVID
The field in the RPL extension that returns the resource identifier of the new conversation. This field is labeled RPL6CNVD in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST.

For half-duplex conversations, this field can have the following values:
X'00' RESET
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOC
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values.
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE_ONLY_LOG
X'86' PENDING_RESET_LOG

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether another FMH-5, other than the one currently being passed to the application program on this APPCCMD, has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No other FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER
X'0030'
PROGRAM_ERROR_NO_TRUNC
X'0034'
PROGRAM_ERROR_PURGING
X'0038'
PROGRAM_ERROR_TRUNC
X'003C'
SERVICE_ERROR_NO_TRUNC
X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

LOGMODE
The field in the RPL extension that returns the logon mode name of the session over which the FMH-5 is being returned on this APPCCMD macroinstruction. It is an 8-byte name, padded on the right with blanks. It is labeled RPL6MODE.

LUNAME
The field in the RPL extension that returns the name of the partner LU that sent the FMH-5 being returned on this APPCCMD macroinstruction. This LU name is the network name of the partner LU. It is an 8-byte name, padded on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID
The field in the RPL extension that returns the network identifier of the partner LU that sent the FMH-5 being returned on this APPCCMD macroinstruction.

This network identifier is the identifier of the partner LU. It can be up to eight characters in length. If it is fewer than eight characters, VTAM pads it on the right with blanks. This field is labeled RPL6NET in the RPL extension.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application the size, in bytes, of the FMH-5. This field is labeled RPLRLEN in the RPL. If the RCPRI field equals X'0000', (OK), RECLEN specifies the number of bytes of the supplied AREA field that were used to return the FMH-5 to the application program.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 4-byte sense code. This sense code has meaning if the RCPRI return code indicates a resource failure problem. It is labeled RPL6SNSI. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated.

SESSID
The field in the RPL extension that, when SESSIDL is not equal to 0, returns a session instance identifier for the session over which the FMH-5 was received. The format of the session instance identifier is described in the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. It is labeled RPL6RSIG. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

SLS
The field in the RPL extension that indicates whether or not the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

WHATRCV
The field in the RPL extension that returns a mask specifying what the application program received. It is labeled RPL6WHAT. The application program should examine this WHATRCV mask only when RCPRI indicates X'0000'. Otherwise, WHATRCV has no meaning.

When RCPRI indicates OK, one or more bits in the mask can be turned *on* (contain a value of B‘1’) to indicate the type of information that has been received. For instance, if the application program is being passed both conversation data and a request for confirmation, both the DATA and CONFIRM bits will be set *on*; the other bits will be set *off*.

The 2-byte WHATRCV mask has the following format.

RPL6RCV1		**RPLRCV2**	
Bit	**Meaning**	**Bit**	**Meaning**
0	DATA	0	PARTIAL_PS_HEADER
1	DATA_COMPLETE	1–7	Reserved
2	DATA_INCOMPLETE		
3	SEND		
4	CONFIRM		
5	DEALLOCATE		
6	LOG_DATA		
7	PS_HEADER		

For example, a WHATRCV value indicating that DATA has been received would be represented by X'8000'. Refer to *z/OS Communications Server: SNA Programmer’s LU 6.2 Guide* for a discussion of the meaning of this field.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:
- VTAM-to-APPL required information vector (X'10')
- Partner’s DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner’s nonce vector (X'14')
- PCID vector (X'17')
- Session information vector (X'19')
- Partner’s application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_NEGATIVE_ RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'008C'	X'0000'	PARTNER_COMMITTED_PROTOCOL_VIOLATION
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE

RCPRI	RCSEC	Meaning
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RCVFMH5, QUALIFY=NULL

Purpose

This macroinstruction receives an FMH-5, which begins the application program's participation in a conversation.

This macroinstruction allows the application to specify how expedited information is received.

Usage

When this macroinstruction is issued, VTAM copies the FMH-5, which represents a new conversation, into the area specified on the AREA parameter. When the macroinstruction completes, the new conversation identifier can be found in the CONVID field. The new conversation will be in RECEIVE state for half-duplex conversations and in SEND/RECEIVE state for full-duplex conversations.

The application program can use the FMH-5 to perform conversation level security processing. Also, the FMH-5 indicates whether any GDS fields, such as DCE security or program initialization (PIP) data, follows the FMH-5. If so, the application program should issue APPCCMD CONTROL=RECEIVE to receive the GDS data.

If no FMH-5 is available for the application to receive, this macroinstruction is rejected with an RCPRI return code of X'0060'.

For information on how the application program is informed that an FMH-5 is ready to be received, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction can be issued from the RESET conversation state.

This macroinstruction is not mode-specific and might be issued for a mode that is retained for persistent LU-LU sessions. However, an FMH-5 is not returned for a mode that is being retained for persistent LU-LU sessions when this macroinstruction is issued.

Syntax

```
►►
┌──────────┐APPCCMD CONTROL=RCVFMH5┌──────────────────┐,RPL=┌─rpl_address_field──────(8)─┐►
└name      ┘                       └,QUALIFY=NULL     ┘     └(rpl_address_register)─────┘

►──────────────────────────────────────────────────────────────────────────────────────────►
│                                                   (1)  │
└,AAREA=┌─rpl_extension_address_field──────────┐
        └(rpl_extension_address_register)──────┘

►──────────────────────────────────────────────────────────────────────────────────────────►
│                                   (1)  │
└,ACB=┌─acb_address_field──────────┐
      └(acb_address_register)──────┘
```

```
►──┬──────────────
```

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the FMH-5. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the size of the supplied buffer area. The supplied buffer area must be large enough to contain the entire FMH-5. An FMH-5 is at most 255 bytes in length (it has only one byte for a length count). If a 255-byte buffer is used to receive the FMH-5, the RCVFMH5 macroinstruction will never fail for lack of buffer space. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running

under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE

Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA

Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS

Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA

Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD

Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS

Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB

Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL

Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register*)
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register*)
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register*)
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CGID
Specifies the 32-bit conversation group identifier.

It is labeled RPL6CGID in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'00' RESET
X'02' RECEIVE
X'08' END_CONVERSATION

For full-duplex conversations, this field can have the following values.
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CONVID
The field in the RPL extension that returns the resource identifier of the new conversation. This field is labeled RPL6CNVD in the RPL extension.

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether another FMH-5, other than the one currently being passed to the application program on this APPCCMD, has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No other FMH-5s are waiting to be received by the application program.

LOGMODE
The field in the RPL extension that returns the logon mode name of the session over which the FMH-5 is being returned on this APPCCMD macroinstruction. It is an 8-byte name, padded on the right with blanks. It is labeled RPL6MODE in the RPL extension.

LUNAME
The field in the RPL extension that returns the name of the partner LU that sent the FMH-5 being returned on this APPCCMD macroinstruction. This LU name is the network name of the partner LU. It is an 8-byte name, padded on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID
The field in the RPL extension that returns the network identifier of the partner LU that sent the FMH-5 being returned on this APPCCMD macroinstruction.

This network identifier is the identifier of the partner LU. It can be up to eight characters in length. If it is fewer than eight characters, VTAM pads it on the right with blanks. This field is labeled RPL6NET in the RPL extension.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application the size, in bytes, of the FMH-5. This field is labeled RPLRLEN in the RPL. If the RCPRI field equals X'0000', (OK), RECLEN specifies the number of bytes of the supplied AREA field that were used to return the FMH-5 to the application program. If the (RCPRI, RCSEC) fields equal X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT, it indicates the size of the FMH-5. However, in the latter case, because the supplied buffer was not large enough to contain the entire FMH-5, the FMH-5 is not returned to the application program. The application program is informed, through the FMH5LEN, of how large the buffer must be in order to receive the FMH-5.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 4-byte sense code. This sense code has meaning if the RCPRI return code indicates a resource failure problem. It is labeled RPL6SNSI in the RPL extension. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated.

SESSID
The field in the RPL extension that, when SESSIDL is not equal to 0, returns a session instance identifier for the session over which the FMH-5 was received. The format of the session instance identifier is described in *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether or not the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- PCID vector (X'17')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is RECEIVE after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_ INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0060'	X'0000'	NO_FMH5_AVAILABLE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A4'	X'0000'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RCVFMH5, QUALIFY=QUEUE

Purpose

This macroinstruction receives an FMH-5, which begins the application program's participation in a conversation.

This macroinstruction allows the application to specify how expedited information is received.

Usage

When this macroinstruction is issued, VTAM copies the FMH-5, which represents a new conversation, into the area specified on the AREA parameter. When the macroinstruction completes, the new conversation identifier can be found in the CONVID field. The new conversation will be in RECEIVE state for half-duplex conversations and in SEND/RECEIVE state for full-duplex conversations.

If this macroinstruction is issued before an FMH-5 is received, VTAM waits for the FMH-5 to complete the macroinstruction. When an FMH-5 is received, VTAM bypasses the ATTN exit. If VTAM receives the FMH-5 before this macroinstruction is issued, VTAM schedules the ATTN exit. In either case, VTAM then moves the FMH-5 to the application's buffer and returns the CONVID and other return parameters. VTAM retains any data that accompanies the FMH-5.

The application program can use the FMH-5 to perform conversation level security processing. Also, the FMH-5 indicates whether any GDS fields, such as DCE security or program initialization (PIP) data, follows the FMH-5. If so, the application program should issue APPCCMD CONTROL=RECEIVE to receive the PIP data.

For information on how the application program is informed that an FMH-5 is ready to be received, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction can be issued from the RESET conversation state.

This macroinstruction is not mode-specific and might be issued for a mode that is retained for persistent LU-LU sessions. However, an FMH-5 is not returned for a mode that is being retained for persistent LU-LU sessions when this macroinstruction is issued.

Syntax

```
►►
┌──────────┐
└name      ┘APPCCMD CONTROL=RCVFMH5,QUALIFY=QUEUE,RPL=┬─rpl_address_field──────────(8)─┬►
                                                      └─(rpl_address_register)─────────┘

►──┬──────────────
```

```
►──┬──────────────
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the FMH-5. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the size of the supplied buffer area. An FMH-5 is, at most, 255 bytes in length. Because the application cannot determine the length of the FMH-5 when the RCVFMH5 request is queued, VTAM fails this macroinstruction if the length of AREALEN is less than 255 with a RCPRI, RCSEC combination of X'002C', X'0008'. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs

running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information on completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RTSRTRN
Specifies the manner in which the Request_To_Send_Received indication is to be reported to the application on subsequent macroinstructions.

RTSRTRN=BOTH
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on all APPCCMDs that return these parameters.

RTSRTRN=EXPD
Specifies that the Request_To_Send_Received indication can be reported in the SIGRCV and SIGDATA fields on an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data to be associated with the new conversation. Whenever an APPCCMD macroinstruction completes for this conversation, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CGID
Specifies the 32-bit conversation group identifier. It is labeled RPL6CGID in the RPL extension.

CONVID
The field in the RPL extension that returns the resource identifier of the new conversation. This field is labeled RPL6CNVD in the RPL extension.

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST.

For half-duplex conversations, this field can have the following values:
X'00' RESET

X'02' RECEIVE
X'08' END_CONVERSATION

For full-duplex conversations, this field can have the following values.
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

CRYPTLVL
Indicates the encryption level for the conversation. This field is labeled RPL6CRYP in the RPL extension.

NONE (B'00')
No data is to be encrypted.

SELECTIVE (B'01')
The application program specifies the data that is to be encrypted.

REQUIRED (B'11')
All data is to be encrypted.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether another FMH-5, other than the one currently being passed to the application program on this APPCCMD, has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No other FMH-5s are waiting to be received by the application program.

LOGMODE
The field in the RPL extension that returns the logon mode name of the session over which the FMH-5 is being returned on this APPCCMD macroinstruction. It is an 8-byte name, padded on the right with blanks. It is labeled RPL6MODE.

LUNAME
The field in the RPL extension that returns the name of the partner LU that sent the FMH-5 being returned on this APPCCMD macroinstruction. This LU name is the network name of the partner LU. It is an 8-byte name, padded on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID
The field in the RPL extension that returns the network identifier of the partner LU that sent the FMH-5 being returned on this APPCCMD macroinstruction.

This network identifier is the identifier of the partner LU. It can be up to eight characters in length. If it is fewer than eight characters, VTAM pads it on the right with blanks. This field is labeled RPL6NET in the RPL extension.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application the size, in bytes, of the FMH-5. This field is labeled RPLRLEN in the RPL. If the RCPRI field equals X'0000', (OK), RECLEN specifies the number of bytes of the supplied AREA field that were used to return the FMH-5 to the application program.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 4-byte sense code. This sense code has meaning if the RCPRI return code indicates a resource failure problem. It is labeled RPL6SNSI. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated.

SESSID
The field in the RPL extension that, when SESSIDL is not equal to 0, returns a session instance identifier for the session over which the FMH-5 was received. The format of the session instance identifier is described in the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*. This field is labeled RPL6SSID in the RPL extension.

SESSIDL
The field in the RPL extension that returns the length of the session instance identifier, which is itself returned in the SESSID field. Values in the range 0–8 are valid. This field is labeled RPL6SIDL in the RPL extension.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

Vectors Returned

VTAM may return the following vectors in the area supplied by the VTRINA parameter:

- VTAM-to-APPL required information vector (X'10')
- Partner's DCE capabilities vector (X'12')
- Local nonce vector (X'13')
- Partner's nonce vector (X'14')
- PCID vector (X'17')
- Session information vector (X'19')
- Partner's application capabilities vector (X'1A')

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is RECEIVE after successful processing.

For full-duplex conversations, the conversation state is SEND/RECEIVE after successful processing.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See"Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_ INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'002E'	VECTOR AREA NOT VALID
X'002C'	X'002F'	VECTOR AREA LENGTH INSUFFICIENT
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A4'	X'0000'	MODE_MUST_BE_RESTORED_BEFORE_USING
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RECEIVE, QUALIFY=ANY

Purpose

This macroinstruction receives normal information on a conversation that is in continue-any mode. Unlike other macroinstructions that are used to receive data, the application program does not specify a partner conversation. Instead, the macroinstruction is associated with the first conversation that is in continue-any mode and that receives data.

Usage

This macroinstruction can be used when the application program is maintaining multiple asynchronous conversations. Instead of issuing APPCCMD CONTROL=RECEIVE, QUAILFY=SPEC|ISPEC for each conversation, the application program can put the conversations in continue-any mode and issue a single APPCCMD CONTROL=RECEIVE, QUALIFY=ANY.

When VTAM receives data on a continue-any mode conversation, VTAM copies the data into the data area that is specified on the AREA parameter and if sufficient data has been received, then VTAM completes the macroinstruction. The conversation identifier of the conversation that is used to complete the macroinstruction is placed in the CONVID field.

This macroinstruction can be used to receive application program data, conversation status information, and confirmation requests. However, it cannot be used to receive error log information. The application program must use APPCCMD CONTROL=RECEIVE, QUAILFY=SPEC|ISPEC to receive error log information.

If VTAM receives notification that a conversation fails on a continue-any mode, this macroinstruction completes with a nonzero return code.

Multiple APPCCMD CONTROL=RECEIVE, QUALIFY=ANY macroinstructions can be outstanding concurrently. The order in which these macroinstructions complete is not necessarily the order in which they were issued. This means that if a conversation is left in continue-any mode, data from multiple RECEIVEs could arrive out of order. If the application program cannot detect this and process the data properly, the application program should specify CONMODE=CS on the APPCCMD CONTROL=RECEIVE, QUALIFY=ANY macroinstruction. For more information on specifying CONMODE, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

APPCCMD CONTROL=RECEIVE, QUALIFY=ANY can be issued when no conversations exist that are in continue-any mode and in RECEIVE, SEND/RECEIVE, or RECEIVE-ONLY state. The APPCCMD waits until one or more conversations are placed into continue-any mode and the right state.

An any-mode RECEIVE can lock out a specific-mode RECEIVE. For example, if an application program has issued an any-mode RECEIVE that receives data in terms of buffers, and enough data has not arrived to satisfy the buffer length, VTAM waits until enough data arrives to satisfy the buffer length before honoring a specific mode RECEIVE for the conversation.

This macroinstruction does not directly correspond to any architected verb described in the LU 6.2 architecture.

Context

Input states are not applicable to this macroinstruction. Only information for a conversation in RECEIVE, SEND/RECEIVE, or RECEIVE_ONLY state and continue-any mode satisfies this type of RECEIVE.

Syntax

```
►►
  [name] APPCCMD CONTROL=RECEIVE ,QUALIFY=ANY(2) ,RPL= rpl_address_field(8)
                                                       (rpl_address_register)
►
  ,AAREA= rpl_extension_address_field(1)
          (rpl_extension_address_register)
►
  ,ACB= acb_address_field(1)
        (acb_address_register)
►
  ,AREA= data_area_address_field(1)
         (data_area_address_register)
►
  ,AREALEN= data_area_length(1)                ,BRANCH= NO(1)
            (data_area_length_register)                YES
►
  ,CD= DEFER(2)                ,CONMODE= BUFFCA(2)          ,CONXMOD= CA(2)
       IMMED                            CS                           CS
                                        LLCA                         SAME
                                        SAME
►
  ,ECB= INTERNAL(3)(1)
        ecb_address_field
        (ecb_address_register)
  ,EXIT= exit_routine_address_field(4)(1)
         (exit_routine_address_register)
```

```
►──┬──────────────────────────────────────────────────────────────────────────►
   │                (7)
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────┬──)─┘
               │          (1)
               ├──ASY──────────────────────────────────────────────────────┤
               │   (1)
               └──SYN──────────────────────────────────────────────────────┘
               │                 (1)   (5)
               ├──KEEPSRB──────────────────────────────────────────────────┤
               │                  (1)   (6)
               └──NKEEPSRB─────────────────────────────────────────────────┘
               │                 (1)
               ├──NBUFFLST─────────────────────────────────────────────────┤
               │                 (1)
               └──XBUFLST──────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────►
   │                                                   (2)
   └─,VTROUTA=──┬─vector_address_field──────────────────────────────────────┘
                └─(vector_address_register)──────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────►◄
   │                                                   (2)
   └─,VTROUTL=──┬─vector_length_field───────────────────────────────────────┘
                └─(vector_length_register)───────────────────────────────────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID

(CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

When OPTCD=XBUFLST, AREA specifies an address in which VTAM is to build an extended buffer list. The AREALEN field of the RPL specifies a length of this area that is a nonzero multiple of 48 bytes. Each entry in the buffer list points to a CSM buffer. For each list entry, VTAM provides the CSM token, data length and information necessary for the application to address the storage (address and data space ALET). Note that a large buffer list area can help prevent excessive API crossings. The format of the extended buffer list pointed to by the AREA parameter is mapped by the ISTBLXEN mapping DSECT.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive.

If OPTCD=XBUFLST, AREALEN specifies the length of the area in which VTAM builds a buffer list. The buffer list in turn points to the data that has been received. The AREALEN parameter specifies an area length that is a nonzero multiple of 48 bytes.

This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CD
Specifies whether the LU immediately goes to SEND or whether the LU defers the SEND transition by going into PENDING_SEND when a change of direction is received with no data.

CD=DEFER
Specifies that the conversation state will be in PENDING_SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CD=IMMED
Specifies that the conversation state will be in SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be

informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NBUFFLST
Specifies that the AREA field contains the address of the area in which the application is to receive the data. The RECLEN field specifies the length of the data area.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. VTAM builds an extended buffer list in the address specified by the AREA parameter. Each entry in the buffer list points to a CSM buffer containing the data being received by the application. The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

Note: Application programs running in TCB-mode supervisor state must specify BRANCH=YES for HPDT requests.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

VTROUTA=*vector_address_field*

VTROUTA=(*vector_address_register***)**
Specifies the address of the area where the application places vector list information for VTAM. If OPTCD=XBUFLST is specified, this field must point to the XBUFLST-receive vector (ISTAPC82), which is mapped by ISTAPCVL. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.)

This field is labeled RPL6VAOA in the RPL extension.

VTROUTL=*vector_length_field*

VTROUTL=(*vector_length_register***)**
Specifies the length of the area where the application places vector list information for VTAM. This field is labeled RPL6VAOL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension. For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

CONVID
Specifies the resource identifier of the conversation on which information was received. This field is labeled RPL6CNVD in the RPL extension.

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC
X'003C'
SERVICE_ERROR_NO_TRUNC
X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of data the application program received up to the maximum. If the application program receives information other than data, this variable is set to 0. When OPTCD=XBUFLST is specified, VTAM returns the actual length of the extended buffer list that is built in the buffer list area pointed to by the AREA operand. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not recognized by VTAM.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

WHATRCV

The field in the RPL extension that returns a mask specifying what the application program received. It is labeled RPL6WHAT in the RPL extension. The application program should examine this WHATRCV mask *only* when RCPRI indicates X'0000'. Otherwise, WHATRCV has no meaning.

When RCPRI indicates OK, one or more bits in the mask can be turned on (contain a value of B'1') to indicate the type of information that has been received. For instance, if the application program is being passed both conversation data and a request for confirmation, both the DATA and CONFIRM bits will be set on; the other bits will be set off.

The 2-byte WHATRCV mask has the following format:

RPL6RCV1

Bit	Meaning
0	DATA
1	DATA_COMPLETE
2	DATA_INCOMPLETE
3	SEND
4	CONFIRM
5	DEALLOCATE
6	LOG_DATA

RPL6RCV2

Bit	Meaning
0	PARTIAL_PS_HEADER
1–7	Reserved

RPL6RCV1		RPL6RCV2	
Bit	**Meaning**	**Bit**	**Meaning**
7	PS_HEADER		

For example, a WHATRCV value indicating that DATA has been received would be represented by X'8000'.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a discussion of the meaning of this field. However, LOG_DATA cannot be set on this macroinstruction.

State Changes

See the description of the WHATRCV mask for a description of the state changes that occur when RCPRI indicates OK.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_ CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_ SUPPORTED_BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_ NO_RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_ RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_ SUPPORTED_BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0030'	PARAMETER_ERROR— STORAGE_TYPE_NOT_VALID
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD

RCPRI	RCSEC	Meaning
X'002C'	X'0033'	PARAMETER_ERROR— A_REQUIRED_VECTOR_WAS_NOT_PROVIDED_ OR_SPECIFIED_INCORRECTLY
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'008C'	X'0000'	PARTNER_COMMITTED_PROTOCOL_VIOLATION
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED

APPCCMD CONTROL=RECEIVE, QUALIFY=IANY

Purpose

This macroinstruction receives normal information that is immediately available on a conversation that is in continue-any mode. VTAM does not wait for data to be received before completing this macroinstruction.

Usage

When this macroinstruction is issued, VTAM copies all data that is immediately available into the supplied data area or control block that is specified by the AREA parameter. VTAM also returns the identification of the conversation that satisfied the macroinstruction in the CONVID parameter.

This macroinstruction can be used to receive application program data, conversation status information, and confirmation requests. However, it cannot be used to receive error log information. The application program must use APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to receive error log information.

This macroinstruction does not directly correspond to any architected verb described in the LU 6.2 architecture.

If no data is immediately available, an RCPRI,RCSEC of (X'0000', X'0008') NO_INFORMATION_IMMEDIATELY_AVAILABLE is returned to the application.

Context

Input states are not applicable to this macroinstruction. Only data for a conversation in RECEIVE, SEND/RECEIVE, or RECEIVE_ONLY state and continue-any mode satisfies this type of RECEIVE.

Syntax

```
►►
 [name] APPCCMD CONTROL=RECEIVE ,QUALIFY=IANY (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
► [,AARREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1)] ►
► [,ACB= [acb_address_field | (acb_address_register)] (1)] ►
► [,AREA= [data_area_address_field | (data_area_address_register)] (1)] ►
► [,AREALEN= [data_area_length | (data_area_length_register)] (1)] [,BRANCH= [NO | YES] (1)] ►
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                    (2)                                                                                                          │
   └─,CD=──┬─DEFER──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────
           └─IMMED──┘                                                                                                                
                                                                                                                                     
   │                                  (2)                                                                                             │
   └─,CONMODE=──┬─BUFFCA──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────
                ├─CS──────┤                                                                                                            
                ├─LLCA────┤                                                                                                            
                └─SAME────┘                                                                                                            
                                                                                                                                     
   │                                  (2)                                                                                             │
   └─,CONXMOD=──┬─CA──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────
                ├─CS──┤                                                                                                                
                └─SAME┘                                                                                                                

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (3)    (1)                                                                      │
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────────
            ├─ecb_address_field──────────┤                                                                                            
            └─(ecb_address_register)─────┘                                                                                            
                                                                                                                                     
                                                                    (4)    (1)                                                      │
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)─────┘                                                                                 

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                 (7)                                                                                                               │
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │          (1)                                                                                                          
               ├─ASY──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │     (1)                                                                                                               
               └─SYN──────┘                                                                                                            
               │               (1)    (5)                                                                                              
               ├─KEEPSRB──────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                 (1)    (6)                                                                                            
               └─NKEEPSRB──────────────────────┘                                                                                     
               │               (1)                                                                                                     
               ├─NBUFFLST─────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                 (1)                                                                                                   
               └─XBUFLST──────────────────────┘                                                                                      
                                                                                                                                     )

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (2)                                                                              │
   └─,VTROUTA=──┬─vector_address_figure──────────────────────────────────────────────────────────────────────────────────────────────
                └─(vector_address_register)─────┘                                                                                     

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                  (2)                                                                              │
   └─,VTROUTL=──┬─vector_length_figure──────────────────────────────────────────────────────────────────────────────────────────────
                └─(vector_length_register)─────┘                                                                                      
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.
5. KEEPSRB is meaningful only for synchronous operations.
6. NKEEPSRB is meaningful only for synchronous operations.
7. You can code more than one suboperand on OPTCD, but no more than one from each group.
8. See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

When OPTCD=XBUFLST, AREA specifies an address in which VTAM is to build an extended buffer list. The AREALEN field of the RPL specifies a length of this area that is a nonzero multiple of 48 bytes. Each entry in the buffer list points to a CSM buffer. For each list entry, VTAM provides the CSM token, data length and information necessary for the application to address the storage (address and data space ALET). Note that a large buffer list area can help prevent excessive API crossings. The format of the extended buffer list pointed to by the AREA parameter is mapped by the ISTBLXEN mapping DSECT.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive.

If OPTCD=XBUFLST, AREALEN specifies the length of the area in which VTAM builds a buffer list. The buffer list in turn points to the data that has been received. The AREALEN parameter specifies an area length that is a nonzero multiple of 48 bytes.

This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CD
Specifies whether the LU immediately goes to SEND or whether the LU defers the SEND transition by going into PENDING_SEND when a change of direction is received with no data.

CD=DEFER
Specifies that the conversation state will be in PENDING_SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CD=IMMED
Specifies that the conversation state will be in SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type

macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NBUFFLST
Specifies that the AREA field contains the address of the area in which the application is to receive the data. The RECLEN field specifies the length of the data area.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. VTAM builds an extended buffer list in the address specified by the AREA parameter. Each entry in the buffer list points to a CSM buffer containing the data being received by the application. The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

Note: Application programs running in TCB-mode supervisor state must specify BRANCH=YES for HPDT requests.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

VTROUTA=*vector_address_field*

VTROUTA=(*vector_address_register***)**
Specifies the address of the area where the application places vector list information for VTAM. If OPTCD=XBUFLST is specified, this field must point to the XBUFLST-receive vector (ISTAPC82), which is mapped by ISTAPCVL. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.)

This field is labeled RPL6VAOA in the RPL extension.

VTROUTL=*vector_length_field*

VTROUTL=(*vector_length_register***)**
Specifies the length of the area where the application places vector list information for VTAM. This field is labeled RPL6VAOL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

CONVID
Specifies the resource identifier of the conversation on which information was received. A value is placed in this field by VTAM only if QUALIFY=*ANY. This field is labeled RPL6CNVD in the RPL extension.

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE,

QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of data the application program received up to the maximum. If the application program receives information other than data, this variable is set to 0. When OPTCD=XBUFLST is specified, VTAM returns the actual length of the extended buffer list that is built in the buffer list area pointed to by the AREA operand. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE

The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not recognized by VTAM.

SIGDATA

The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')

A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')

No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

WHATRCV

The field in the RPL extension that indicates what the application program received. It is labeled RPL6WHAT in the RPL extension. The application

program should examine the WHATRCV field *only* when RCPRI indicates X'0000'. Otherwise, WHATRCV has no meaning.

When RCPRI indicates OK, one or more bits in the mask can be turned on (contain a value of B'1') to indicate the type of information that has been received. For instance, if the application program is being passed both conversation data and a request for confirmation, both the DATA and CONFIRM bits will be set on; the other bits will be set off.

The 2-byte WHATRCV mask has the following format:

RPL6RCV1

Bit	Meaning
0	DATA
1	DATA_COMPLETE
2	DATA_INCOMPLETE
3	SEND
4	CONFIRM
5	DEALLOCATE
6	LOG_DATA
7	PS_HEADER

RPL6RCV2

Bit	Meaning
0	PARTIAL_PS_HEADER
1–7	Reserved

For example, a WHATRCV value indicating that DATA has been received would be represented by X'8000'.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a discussion of the meaning of this field. However, LOG_DATA cannot be set on this macroinstruction.

State Changes

See the description of the WHATRCV mask for a description of the state changes that occur when RCPRI indicates OK.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0008'	NO_IMMEDIATELY_AVAILABLE_INFORMATION
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY

RCPRI	RCSEC	Meaning
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR— SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ADDRESS_ SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_NON-APPC
X'002C'	X'0030'	PARAMETER_ERROR— STORAGE_TYPE_NOT_VALID
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'002C'	X'0033'	PARAMETER_ERROR— A_REQUIRED_VECTOR_WAS_NOT_PROVIDED_ OR_SPECIFIED_INCORRECTLY
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_NEGATIVE_ RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_NEGATIVE_ RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'008C'	X'0000'	PARTNER_COMMITTED_PROTOCOL_VIOLATION
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE.
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED

APPCCMD CONTROL=RECEIVE, QUALIFY=ISPEC

Purpose

This macroinstruction receives normal information that is immediately available from a specified conversation. The conversation may be in continue-any or continue-specific mode. VTAM does not wait for more data to be received before completing this macroinstruction.

Usage

When this macroinstruction is issued, VTAM copies all data that is immediately available into the supplied data area or control block that is specified by the AREA parameter. The AREALEN parameter specifies the length of the data area. VTAM does not wait to receive any more data before completing the macroinstruction request. If there is no information available, VTAM issues an RCPRI, RCSEC combination of X'0000', X'0008', NO_IMMEDIATELY_AVAILABLE_INFORMATION.

When this macroinstruction completes, the RECLEN field indicates how much data was written to the data area. The WHATRCV field indicates what type of data was received.

If VTAM is processing APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY for a conversation and the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=ISPEC for the same conversation, the QUALIFY=ISPEC request fails with an RCPRI, RCSEC combination of X'0000', X'0003', RECEIVE_SPECIFIC_REJECTED. VTAM cannot allow a specific-mode RECEIVE while an any-mode RECEIVE is being processed.

This macroinstruction corresponds to the RECEIVE_IMMEDIATE verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- RECEIVE
- PEND_END_CONV_LOG
- PEND_RCV_LOG

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND/RECEIVE
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
┌──────────┐
└name      ┘APPCCMD CONTROL=RECEIVE──,QUALIFY=ISPEC──────(2)──,RPL=──┬─rpl_address_field──────────┬──(8)──►
                                                                      └─(rpl_address_register)─────┘
```

```
►──┬──────────────
```

```
►──┬──────────────
```

excessive API crossings. The format of the extended buffer list pointed to by the AREA parameter is mapped by the ISTBLXEN mapping DSECT.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive.

If OPTCD=XBUFLST, AREALEN specifies the length of the area in which VTAM builds a buffer list. The buffer list in turn points to the data that has been received. The AREALEN parameter specifies an area length that is a nonzero multiple of 48 bytes.

This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CD
Specifies whether the LU immediately goes to SEND or whether the LU defers the SEND transition by going into PEND_SEND when a change of direction is received with no data.

CD=DEFER
Specifies that the conversation state will be in PEND_SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CD=IMMED
Specifies that the conversation state will be in SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an

asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

FILL
Specifies whether the application program is to receive data in terms of the logical-record format of the data. This parameter corresponds to FILL=LL|BUFFER described in the LU 6.2 architecture. This field is labeled RPL6FILL in the RPL extension.

FILL=BUFF
Specifies the application program is to receive data independently of its logical-record format, up to the length specified by the AREALEN field of the RPL. FILL=BUFF corresponds to FILL=BUFFER on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

FILL=LL
Specifies the application program is to receive one logical record, or whatever portion of the logical record is available, up to the length specified by the AREALEN field of the RPL. FILL=LL corresponds to FILL=LL on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NBUFFLST
Specifies that the AREA field contains the address of the area in which the application is to receive the data. The RECLEN field specifies the length of the data area.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. VTAM builds an extended buffer list in the address specified by the AREA parameter. Each entry in the buffer list points to a CSM buffer containing the data being received by the application. The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

Note: Application programs running in TCB-mode supervisor state must specify BRANCH=YES for HPDT requests.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

VTROUTA=*vector_address_field*

VTROUTA=(*vector_address_register***)**
Specifies the address of the area where the application places vector list information for VTAM. If OPTCD=XBUFLST is specified, this field must point to the XBUFLST-receive vector (ISTAPC82), which is mapped by ISTAPCVL. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.)

This field is labeled RPL6VAOA in the RPL extension.

VTROUTL=*vector_length_field*

VTROUTL=(*vector_length_register***)**
Specifies the length of the area where the application places vector list information for VTAM. This field is labeled RPL6VAOL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have t the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC
X'0034'
PROGRAM_ERROR_PURGING
X'0038'
PROGRAM_ERROR_TRUNC
X'003C'
SERVICE_ERROR_NO_TRUNC
X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'0048'
RESOURCE_FAILURE,_NO_RETRY
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of data the application program received. If the application program receives information other than data, this variable is set to 0. When OPTCD=XBUFLST is specified, VTAM returns the actual length of the extended buffer list that is built in the buffer list area pointed to by the AREA operand. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA
The field in the RPL extension in which the signal code and signal extension

fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

- X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')

A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')

No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

WHATRCV

The field in the RPL extension that indicates what the application program received. It is labeled RPL6WHAT in the RPL extension. The application program should examine the WHATRCV field only when RCPRI indicates X'0000'. Otherwise, WHATRCV has no meaning.

When RCPRI indicates OK, one or more bits in the mask can be turned *on* (contain a value of B'1') to indicate the type of information that has been received. For instance, if the application program is being passed both conversation data and a request for confirmation, both the DATA and CONFIRM bits will be set *on*; the other bits will be set *off*.

The 2-byte WHATRCV mask has the format shown in Table 1.

Table 1. Format of WHATRCV Mask

RPL6RCV1		RPL6RCV2	
Bit	**Meaning**	**Bit**	**Meaning**
0	DATA	0	PARTIAL_PS_HEADER
1	DATA_COMPLETE	1–7	Reserved
2	DATA_INCOMPLETE		
3	SEND		
4	CONFIRM		
5	DEALLOCATE		
6	LOG_DATA		
7	PS_HEADER		

For example, a WHATRCV value indicating that DATA has been received would be represented by X'8000'. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a discussion of the meaning of this field.

State Changes

See the description of the WHATRCV mask for state changes when RCPRI indicates OK.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0003'	RECEIVE_SPECIFIC_REJECTED
X'0000'	X'0008'	NO_IMMEDIATELY_AVAILABLE_INFORMATION
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0002'	PARAMETER_ERROR_INVALID_CONVERSATION_ID
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD

RCPRI	RCSEC	Meaning
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ADDRESS_ SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_NON-APPC
X'002C'	X'0030'	PARAMETER_ERROR— STORAGE_TYPE_NOT_VALID
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'002C'	X'0033'	PARAMETER_ERROR— A_REQUIRED_VECTOR_WAS_NOT_PROVIDED_ OR_SPECIFIED_INCORRECTLY
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_NEGATIVE_ RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_NEGATIVE_ RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'008C'	X'0000'	PARTNER_COMMITTED_PROTOCOL_VIOLATION
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED

APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC

Purpose

This macroinstruction receives information on a specified conversation. The conversation may be in any continuation mode.

Usage

When this macroinstruction is issued, VTAM copies any available data from the conversation that is specified by the CONVID parameter to the data area that is specified by the AREA parameter. The AREALEN parameter specifies the length of the data area. If no data is ready to be received on the conversation, VTAM queues the macroinstruction until data arrives.

When this macroinstruction completes, the RECLEN field indicates how much data was written to the data area. The WHATRCV field indicates what type of data was received.

The application program can issue this macroinstruction when the conversation is in SEND state. In this case, VTAM flushes its SEND buffer, sending all buffered information, along with the SEND indicator, to the partner LU. This changes the conversation to RECEIVE state. VTAM then waits for information to arrive. The remote application program can send data to the local application program after it receives the SEND indication.

If VTAM is processing APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY for a conversation and the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC for the same conversation, the QUALIFY=SPEC request fails with an RCPRI, RCSEC return code of X'0000', X'0003'. (VTAM cannot allow a specific-mode RECEIVE while an any-mode RECEIVE is being processed because if a SEND indication was received on the any-mode RECEIVE while the specific-mode RECEIVE was being processed, a SEND indicator would erroneously be sent to the partner LU as a result of the specific-mode RECEIVE.)

This macroinstruction corresponds to the RECEIVE_AND_WAIT verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- RECEIVE
- SEND
- PEND_END_CONV_LOG
- PEND_RCV_LOG

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND/RECEIVE
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
└─name─┘─APPCCMD CONTROL=RECEIVE──,QUALIFY=SPEC──(2)──,RPL=──rpl_address_field──(8)──►
                                                          └─(rpl_address_register)─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,AAREA=──rpl_extension_address_field──(1)──┬──────────────────────────────────┘
             └─(rpl_extension_address_register)─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,ACB=──acb_address_field──(1)──┬──────────────────────────────────┘
           └─(acb_address_register)─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,AREA=──data_area_address_field──(1)──┬──────────────────────────────────┘
            └─(data_area_address_register)─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,AREALEN=──data_area_length──(1)──┬──────────────────────────────────┘
               └─(data_area_length_register)─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,BRANCH=──┬─NO──┬──(1)──┘
              └─YES─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,CD=──┬─DEFER─┬──(2)──┘
          └─IMMED─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,CONMODE=──┬─BUFFCA─┬──(2)──┘
               ├─CS─────┤
               ├─LLCA───┤
               └─SAME───┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,CONVID=──32-bit_resource_id_field──(2)──┬──────────────────────────────────┘
              └─(32-bit_resource_id_register)─┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  └─,CONXMOD=──┬─CA────┬──(2)──┘
               ├─CS────┤
               └─SAME──┘

►─┬──────────────────────────────────────────────────────────────────────────────────►
  │ ┌──────────────────────────────────────────────────────────────────────────────┐
  │ │ ┌──────────────────────────────────────────────────────────────────────────┐
  │ │ │ ┌──────────────────────────────────────────────────────────────────────┐
  │ │ │ │ ┌──────────────────────────────────────────────────────────────────┐
  │ │ │ │ │ ┌──────────────────────────────────────────────────────────────┐
  │ │ │ │ │ │ ┌──────────────────────────────────────────────────────────┐
  │ │ │ │ │ │ │ ┌──────────────────────────────────────────────────────┐
  │ │ │ │ │ │ │ │ ┌──────────────────────────────────────────────────┐
  │ │ │ │ │ │ │ │ │ ┌──────────────────────────────────────────────┐
  │ │ │ │ │ │ │ │ │ │ ┌──────────────────────────────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ ┌──────────────────────────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────────────────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────────────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌────┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ ┌──┐
  │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │ │
```

```
►──┬──────────────────────────────────────────────────────────────────────────►
   │                                   (2)
   └─,VTROUTA=──┬─vector_address_field──────────┬──────────────────────────────
                └─(vector_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────►◄
   │                                   (2)
   └─,VTROUTL=──┬─vector_length_field──────────┬──────────────────────────────
                └─(vector_length_register)─────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

When OPTCD=XBUFLST, AREA specifies an address in which VTAM is to build an extended buffer list. The AREALEN field of the RPL specifies a length of this

area that is a nonzero multiple of 48 bytes. Each entry in the buffer list points to a CSM buffer. For each list entry, VTAM provides the CSM token, data length and information necessary for the application to address the storage (address and data space ALET). Note that a large buffer list area can help prevent excessive API crossings. The format of the extended buffer list pointed to by the AREA parameter is mapped by the ISTBLXEN mapping DSECT.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive.

If OPTCD=XBUFLST, AREALEN specifies the length of the area in which VTAM builds a buffer list. The buffer list in turn points to the data that has been received. The AREALEN parameter specifies an area length that is a nonzero multiple of 48 bytes.

This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CD
Specifies whether the LU immediately goes to SEND or whether the LU defers the SEND transition by going into PEND_SEND when a change of direction is received with no data.

CD=DEFER
Specifies that the conversation state will be in PEND_SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CD=IMMED
Specifies that the conversation state will be in SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA

corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

FILL
Specifies whether the application program is to receive data in terms of the logical-record format of the data. This parameter corresponds to FILL=LL|BUFFER described in the LU 6.2 architecture. This field is labeled RPL6FILL in the RPL extension.

FILL=BUFF
Specifies the application program is to receive data independently of its logical-record format, up to the length specified by the AREALEN field of the RPL. FILL=BUFF corresponds to FILL=BUFFER on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

FILL=LL
Specifies the application program is to receive one logical record, or whatever portion of the logical record is available, up to the length specified by the AREALEN field of the RPL. FILL=LL corresponds to FILL=LL on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NBUFFLST
Specifies that the AREA field contains the address of the area in which the application is to receive the data. The RECLEN field specifies the length of the data area.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. VTAM builds an extended buffer list in the address specified by the AREA parameter. Each entry in the buffer list points to a CSM buffer containing the data being received by the application. The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

Note: Application programs running in TCB-mode supervisor state must specify BRANCH=YES for HPDT requests.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

VTROUTA=*vector_address_field*

VTROUTA=(*vector_address_register***)**
Specifies the address of the area where the application places vector list information for VTAM. If OPTCD=XBUFLST is specified, this field must point to the XBUFLST-receive vector (ISTAPC82), which is mapped by ISTAPCVL. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.)

This field is labeled RPL6VAOA in the RPL extension.

VTROUTL=*vector_length_field*

VTROUTL=(*vector_length_register***)**
Specifies the length of the area where the application places vector list information for VTAM. This field is labeled RPL6VAOL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY

X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:
X'0004'
ALLOCATION_ERROR
X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE
X'001C'
DEALLOCATE_ABEND_TIMER
X'0030'
PROGRAM_ERROR_NO_TRUNC
X'0034'
PROGRAM_ERROR_PURGING
X'0038'
PROGRAM_ERROR_TRUNC
X'003C'
SERVICE_ERROR_NO_TRUNC
X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'0048'
RESOURCE_FAILURE,_NO_RETRY
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual amount of data the application program received. If the application program receives information other than data, this variable is set to 0. When OPTCD=XBUFLST is specified, VTAM returns the actual length of the extended buffer list that is built in the buffer list area pointed to by the AREA operand. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA

The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPL extension.

YES (B'1')

A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')

No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

WHATRCV

The field in the RPL extension that returns a mask specifying what the application program received. It is labeled RPL6WHAT in the RPL. The application program should examine this WHATRCV mask only when RCPRI indicates X'0000'. Otherwise, WHATRCV has no meaning.

When RCPRI indicates OK, one or more bits in the mask can be turned *on* (contain a value of B'1') to indicate the type of information that has been received. For instance, if the application program is being passed both conversation data and a request for confirmation, both the DATA and CONFIRM bits will be set *on*; the other bits will be set *off*.

The 2-byte WHATRCV mask has the following format.

RPL6RCV1		RPL6RCV2	
Bit	**Meaning**	**Bit**	**Meaning**
0	DATA	0	PARTIAL_PS_HEADER
1	DATA_COMPLETE	1–7	Reserved
2	DATA_INCOMPLETE		
3	SEND		
4	CONFIRM		
5	DEALLOCATE		
6	LOG_DATA		
7	PS_HEADER		

For example, a WHATRCV value indicating that DATA has been received would be represented by X'8000'. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a discussion of the meaning of this field.

State Changes

See the description of the WHATRCV mask for state changes when RCPRI indicates OK.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See"Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0003'	RECEIVE_SPECIFIC_REJECTED
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR— SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD

RCPRI	RCSEC	Meaning
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0030'	PARAMETER_ERROR— STORAGE_TYPE_NOT_VALID
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'002C'	X'0033'	PARAMETER_ERROR— A_REQUIRED_VECTOR_WAS_NOT_PROVIDED_ OR_SPECIFIED_INCORRECTLY
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_NEGATIVE_ RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'008C'	X'0000'	PARTNER_COMMITTED_PROTOCOL_VIOLATION
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED

APPCCMD CONTROL=REJECT, QUALIFY=CONV

Purpose

This macroinstruction deallocates a conversation abnormally as well as its underlying session when the application program detects a protocol violation on the conversation.

If the conversation is no longer associated with a session when APPCCMD CONTROL=REJECT, QUALIFY=CONV is issued, VTAM does not unbind the session.

Usage

When the application program detects a protocol violation on the conversation, it issues this macroinstruction and specifies a sense code on the SENSE parameter. VTAM deallocates the conversation first. If the conversation is still associated with a session, VTAM deactivates the session by issuing an UNBIND of type X'FE', which contains the user-specified sense code. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a list of valid UNBIND sense codes.

As an example, suppose the local application program issues an APPCCMD macroinstruction that completes with a return code of PROGRAM_ERROR_NO_TRUNC and LOGRCV=YES, which indicates that an error is detected and that the partner LU is sending error log data. Also, suppose the local application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC to receive the error log data and either no data is received or the data that is received is not error-log data. This means that the partner LU committed a protocol violation, and the application program could issue this macroinstruction to end the conversation and session.

APPCCMD CONTROL=REJECT, QUALIFY=CONV can be issued to cancel an APPCCMD macroinstruction that was issued on the conversation previously. However, it cannot cancel an APPCCMD CONTROL=OPRCNTL, APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY, or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY macroinstruction that has not been matched to a conversation. Nor can it cancel an APPCCMD CONTROL=REJECT, QUALIFY=CONV macroinstruction that was issued previously for the same conversation or an APPCCMD CONTROL=TESTSTAT, QUALIFY=ALL|IALL.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PENDING_DEALLOCATE
- PEND_END_CONV_LOG
- PENDING_SEND
- PEND_RCV_LOG

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:

- SEND/RECEIVE

- RECEIVE_ONLY
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=REJECT ,QUALIFY=CONV ,RPL= rpl_address_field (8)
                                                   (rpl_address_register)
►─────────────────────────────────────────────────────────────────────────►
 ,AAREA= rpl_extension_address_field (1)
         (rpl_extension_address_register)
►─────────────────────────────────────────────────────────────────────────►
 ,ACB= acb_address_field (1)          ,BRANCH= NO (1)
       (acb_address_register)                 YES
►─────────────────────────────────────────────────────────────────────────►
 ,CONVID= 32-bit_resource_id_field (2)
          (32-bit_resource_id_register)
►─────────────────────────────────────────────────────────────────────────►
 ,ECB= INTERNAL (3) (1)
       ecb_address_field
       (ecb_address_register)
 ,EXIT= exit_routine_address_field (4) (1)
        (exit_routine_address_register)
►─────────────────────────────────────────────────────────────────────────►
 ,OPTCD=( ,(7) )
          ASY (1)
              SYN (1)
          KEEPSRB (1) (5)
          NKEEPSRB (1) (6)
►─────────────────────────────────────────────────────────────────────────►◄
 ,SENSE= 32-bit_unbind_sense_code (2)
         (32-bit_unbind_sense_code_register)
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register*)
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing the APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource. The application program is allowed to issue APPCCMDs against other conversations.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register*)
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SENSE=*32-bit_unbind_sense_code*

SENSE=(*32-bit_unbind_sense_code_register*)
Indicates the reason for the APPCCMD CONTROL=REJECT macroinstruction. This field specifies a 32-bit UNBIND (X'FE') sense code. VTAM generates an UNBIND (X'FE') carrying the supplied sense code and ends the conversation. This field is labeled RPL6SNSO in the RPL extension. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information on sense codes.)

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following value:
X'08' END_CONVERSATION

For full-duplex conversations, this field can have the following value:
X'80' FDX_RESET

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return

code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is END_CONV after successful processing.

For full-duplex conversations, the conversations state is FDX_RESET after successful processing.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_ INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0020'	PARAMETER_ERROR—PREVIOUS_REJECT_REQUEST_ OUTSTANDING
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_ SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=REJECT, QUALIFY=CONVGRP

Purpose

This macroinstruction deactivates the session associated with the conversation group and any conversations matched to the session. The application program specifies, through the deactivation type code, that either a protocol violation has occurred or cleanup is necessary.

Usage

If the application program detects a protocol violation committed by the partner LU or if the architected processing indicates that a cleanup deactivation of the session should occur, the application program issues APPCCMD CONTROL=REJECT, QUALIFY=CONVGRP to terminate the session. This session can have an active conversation associated with it. If so, the conversation fails with an indication of an abnormal termination.

By using the deactivation type (DEACTYP) parameter, the application program can indicate that VTAM should send either an UNBIND PROTOCOL_ VIOLATION (X'FE') or an UNBIND CLEANUP (X'0F') to deactivate the session. If the deactivation type parameter is omitted, or is equal to a value other than X'0F' or X'FE', VTAM generates an UNBIND (X'0F'). The sense code parameter is valid only if UNBIND (X'FE') is specified.

The application program must specify the conversation group that is to be deactivated. To do this, it uses the CGID parameters to specify the conversation group identifier.

VTAM posts the application program with successful return codes if no session is active with the specified conversation group identifier.

APPCCMD CONTROL=REJECT, QUALIFY=CONVGRP can be issued without knowledge of any conversations associated with the specified session through the CGID parameter. It corresponds to the DEACTIVATE_CONVERSATION_GROUP verb in the LU 6.2 architecture.

Context

This macroinstruction is not conversation-specific and therefore is not conversation-state-driven.

Syntax

```
►►
┌──────┐
└name  ┘──APPCCMD CONTROL=REJECT──,QUALIFY=CONVGRP──,RPL=──┬─rpl_address_field──────────(8)─┬──►
                                                           └─(rpl_address_register)─────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                   (1)                    │
   └──,AAREA=──┬─rpl_extension_address_field──────────────────────────┬──────────────────────┘
               └─(rpl_extension_address_register)─────────────────────┘
```

```
►──┬──────────────
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CGID=*32-bit_conversation_group_id_field*

CGID=(*32-bit_conversation_group_id_register***)**
Specifies the 32-bit conversation group ID.

This value can be obtained from a previous APPCCMD CONTROL=ALLOC, CONTROL=PREALLOC, or CONTROL=RCVFMH5 macroinstruction. If the CGID operand is not specified, VTAM uses the conversation group ID that is already in the RPL6CGID field on the RPL extension.

The conversation group ID identifies a specific session between two specific LUs. It provides a means by which a VTAM LU 6.2 application program and its partner LU can share serially the same session.

DEACTYP=*8-bit_unbind_type_code*

DEACTYP=(*8-bit_unbind_type_code_register***)**
The UNBIND type code can be specified as cleanup (X'0F') or as protocol violation (X'FE'). If DEACTYP specifies cleanup, the value specified on the SENSE operand will be ignored. However, if DEACTYP specifies protocol error, the UNBIND will flow with the sense code specified by the SENSE operand. If the DEACTYP operand is omitted or a value other than X'0F' or X'FE' is entered, VTAM will generate an UNBIND of X'0F'. The application program can

be posted with a return code of INVALID_DEACTIVATION_TYPE_CODE, but the session may still have been deactivated successfully. This field is labeled RPL6DETP in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SENSE=*32-bit_unbind_sense_code*

SENSE=(*32-bit_unbind_sense_code_register*)
Indicates the reason for the APPCCMD CONTROL=REJECT macroinstruction. This field specifies a 32-bit UNBIND (X'FE') sense code. VTAM generates an UNBIND (X'FE') carrying the supplied sense code and ends the conversation. This field is labeled RPL6SNSO in the RPL extension. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information on sense codes.)

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD

CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

Conversation states do not apply to this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0020'	PARAMETER_ERROR—PREVIOUS_REJECT_REQUEST_ OUTSTANDING
X'002C'	X'0027'	PARAMETER_ERROR—INVALID_DEACTIVATION_TYPE_ CODE
X'002C'	X'002A'	PARAMETER_ERROR—INVALID_CGID_VALUE_SPECIFIED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=REJECT, QUALIFY=SESSION

Purpose

This macroinstruction deactivates the session and any conversation matched to this session. The application program specifies, through the deactivation type code, that either a protocol violation has occurred or cleanup is necessary.

Usage

If the application program detects a protocol violation committed by the partner LU or if the architected processing indicates that a cleanup deactivation of the session should occur, the application program issues APPCCMD CONTROL=REJECT, QUALIFY=SESSION to terminate the session. This session can have an active conversation associated with it. If so, the conversation fails with an indication of an abnormal termination. The application must issue an APPCCMD to receive the conversation failure notification and cause conversation cleanup.

By using the deactivation type (DEACTYP) parameter, the application program can indicate that VTAM should send either an UNBIND PROTOCOL_VIOLATION (X'FE') or an UNBIND CLEANUP (X'0F') to deactivate the session. If the deactivation type parameter is omitted, or is equal to a value other than X'0F' or X'FE', VTAM generates an UNBIND (X'0F'). The sense code parameter is ignored unless an UNBIND (X'FE') is specified.

The application program must specify the session that is to be deactivated. To do this, it uses the SESSID and SESSIDL parameters to specify the session instance identifier. These parameters were made available to the conversation at conversation allocation from the APPCCMD CONTROL=RCVFMH5 macroinstruction and the APPCCMD CONTROL=ALLOC macroinstruction.

VTAM posts the application program with successful return codes if no session is active with the specified session identifier and session identifier length.

APPCCMD CONTROL=REJECT, QUALIFY=SESSION can be issued without knowledge of any conversations associated with the specified session through the SESSID parameter.

Context

This macroinstruction is not conversation-specific and therefore is not conversation-state-driven.

Syntax

```
►►
┌──────────┐
└name      ┘──APPCCMD CONTROL=REJECT──,QUALIFY=SESSION──,RPL=──┬─rpl_address_field──────────┬──►
                                                                │                (8)         │
                                                                └─(rpl_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                  (1)                                     │
   └─,AAREA=──┬─rpl_extension_address_field──────────┬──────────────────────────────────────┘
              │                                      │
              └─(rpl_extension_address_register)─────┘
```

```
►──┬──────────────
```

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

DEACTYP=*8-bit_unbind_type_code*

DEACTYP=(*8-bit_unbind_type_code_register***)**
The UNBIND type code can be specified as cleanup (X'0F') or as protocol violation (X'FE'). If DEACTYP specifies cleanup, the value specified on the SENSE operand will be ignored. However, if DEACTYP specifies protocol error, the UNBIND will flow with the sense code specified by the SENSE operand. If the DEACTYP operand is omitted or a value other than X'0F' or X'FE' is entered, VTAM will generate an UNBIND of X'0F'. The application program can be posted with a return code of INVALID_DEACTIVATION_TYPE_CODE, but the session may still have been deactivated successfully. This field is labeled RPL6DETP in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot

specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SENSE=*32-bit_unbind_sense_code*

SENSE=(*32-bit_unbind_sense_code_register***)**
Indicates the reason for the APPCCMD CONTROL=REJECT macroinstruction. This field specifies a 32-bit UNBIND (X'FE') sense code. VTAM generates an UNBIND (X'FE') carrying the supplied sense code and ends the conversation.

This field is labeled RPL6SNSO in the RPL extension. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information on sense codes.)

SESSID=*session_instance_id_field*

SESSID=(*session_instance_id_register***)**
Specifies the session to be deactivated. The session instance identifier must refer to an active session. (A session must be activated before it can be deactivated.) The session instance identifier is passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction. A session that is in pending activation state cannot be specified. A conversation that is matched to this session fails with a session outage notification. This field is labeled RPL6SSID in the RPL extension.

SESSIDL=*session_instance_id_length*

SESSIDL=(*session_instance_id_length_register***)**
Specifies the length of the session instance ID. The value specified must be greater than 0 and less than or equal to 8. The session instance ID length was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction. This field is labeled RPL6SIDL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

State Changes

Conversation states do not apply to this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0020'	PARAMETER_ERROR—PREVIOUS_REJECT_REQUEST_ OUTSTANDING
X'002C'	X'0023'	PARAMETER_ERROR—INVALID_SESSION_INSTANCE_ IDENTIFIER
X'002C'	X'0027'	PARAMETER_ERROR—INVALID_DEACTIVATION_TYPE_ CODE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=RESETRCV

Purpose

This macroinstruction changes the existing continuation modes of a conversation. For example, it can change the conversation from continue-specific (CS) mode to logical-record-continue-any (LLCA) mode for receiving normal information.

This macroinstruction can also change the existing mode for receiving expedited information.

Usage

When this macroinstruction is issued, VTAM changes the continuation mode for receiving normal information of the conversation specified with the CONVID parameter to the continuation mode specified on the CONMODE parameter.

VTAM also changes the expedited information mode of the conversation specified with the CONXMOD parameter to the expedited information mode specified on the CONXMOD parameter.

For a complete discussion of continuation modes and an example of how this macroinstruction can be used, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- SEND
- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PEND_END_CONV_LOG
- PEND_RCV_LOG

For full-duplex conversation, this macroinstruction can be issued form the following conversations states:

- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►                                                                    (8)
──┬──────────┬──APPCCMD CONTROL=RESETRCV──┬──────────────────┬──,RPL=──┬─rpl_address_field──────────┬──►
  └─name─────┘                            └─,QUALIFY=NULL────┘         └─(rpl_address_register)────┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (1)
   └─,AAREA=──┬─rpl_extension_address_field──────────────────────────┬──
              └─(rpl_extension_address_register)──────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (1)
   └─,ACB=────┬─acb_address_field──────────────────────────────────┬──
              └─(acb_address_register)──────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (1)
   └─,BRANCH=─┬─NO──┬──
              └─YES─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (2)
   └─,CONMODE=─┬─BUFFCA─┬──
               ├─CS─────┤
               ├─LLCA───┤
               └─SAME───┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (2)
   └─,CONVID=─┬─32-bit_resource_id_field──────────────────────────┬──
              └─(32-bit_resource_id_register)──────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (2)
   └─,CONXMOD=─┬─CA─┬──
               ├─CS─┤
               └─SAME─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                  (3)   (1)
   ├─,ECB=────┬─INTERNAL──────────────────────────────────────────┬──
   │          ├─ecb_address_field──────────────────────────────────┤
   │          └─(ecb_address_register)──────────────────────────────┘
   │                                                                  (4)   (1)
   └─,EXIT=───┬─exit_routine_address_field────────────────────────┬──
              └─(exit_routine_address_register)────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                  ┌──────────────────────────┐
   │                                  │                          ,
   │                                  ▼   (7)                    │
   └─,OPTCD=(──┬──────────────────────────────────────────────────┬──)──┘
               │                 (1)
               ├─ASY──────────────────────────────────────────────┤
               │      (1)
               └─SYN──────────────────────────────────────────────┘
               │                 (1)   (5)
               ├─KEEPSRB──────────────────────────────────────────┤
               │                 (1)   (6)
               └─NKEEPSRB─────────────────────────────────────────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-

any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing this APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource (that processes on the SEND/RECEIVE queue if the conversation is half-duplex, or on the SEND queue if the conversation is full-duplex) until the command has completed. The exceptions to this are the APPCCMD CONTROL=REJECT, QUALIFY=CONV macroinstruction and the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstruction. The application program can issue APPCCMDs against the same conversation resource that processes on the RECEIVE (if the conversation is full-duplex), EXPEDITED SEND, EXPEDITED RECEIVE, and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing the APPCCMD on a full-duplex conversation.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM

X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can contain the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

There are no state changes associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SEND, QUALIFY=CONFIRM

Purpose

This macroinstruction sends a confirmation request on a half-duplex conversation to a remote application program and waits for a confirmation reply (either synchronously or asynchronously).

Usage

This macroinstruction can only be used for half-duplex conversations.

When this macroinstruction is issued, VTAM flushes the SEND buffer of the specified conversation and sends a confirmation request. This macroinstruction completes only after the partner LU receives the confirmation request and issues APPCCMD CONTROL=SEND, QUALIFY=CONFRMD.

This macroinstruction enables the local and remote application programs to synchronize their processing with one another. The application program can use this APPCCMD for various transaction program-level functions. For example:

- The application program can issue this APPCCMD immediately following an APPCCMD CONTROL=ALLOC macroinstruction in order to determine whether the allocation of the conversation is successful before sending any data.
- The application program can issue this APPCCMD as a request for acknowledgment of data that it sent to the remote program.

The application program must ensure that APPCCMD CONTROL=SEND, QUALIFY=CONFIRM is not issued by a transaction program against a conversation that was allocated with a synchronization level of NONE.

This macroinstruction corresponds to the CONFIRM verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from following conversation states:

- SEND
- PENDING_SEND

This macroinstruction is not allowed on full-duplex conversations.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
  [name] APPCCMD CONTROL=SEND ,QUALIFY=CONFIRM (2) ,RPL= [rpl_address_field (8)
                                                         (rpl_address_register)] ►
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,AARREA=──┬─rpl_extension_address_field──────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               └─(rpl_extension_address_register)──────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,ACB=──┬─acb_address_field──────────────────────────────────────────────────────────────────────────────────────────────────
            │                                                                                                                       
            └─(acb_address_register)──────────────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,BRANCH=──┬─NO──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               └─YES─────────────────────────────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,CONMODE=──┬─BUFFCA──────────────────────────────────────────────────────────────────────────────────────────────────────────
                │                                                                                                                    
                ├─CS────────────────────────────────────────────────────────────────────────────────────────────────────────────
                │                                                                                                                    
                ├─LLCA──────────────────────────────────────────────────────────────────────────────────────────────────────────
                │                                                                                                                    
                └─SAME──────────────────────────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,CONVID=──┬─32-bit_resource_id_field──────────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               └─(32-bit_resource_id_register)─────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,CONXMOD=──┬─CA──────────────────────────────────────────────────────────────────────────────────────────────────────────────
                │                                                                                                                    
                ├─CS────────────────────────────────────────────────────────────────────────────────────────────────────────────
                │                                                                                                                    
                └─SAME──────────────────────────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────
            │                                                                                                                       
            ├─ecb_address_field──────────────────────────────────────────────────────────────────────────────────────────────────
            │                                                                                                                       
            └─(ecb_address_register)─────────────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────────
             │                                                                                                                      
             └─(exit_routine_address_register)──────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                                                                                                
   └─,OPTCD=(──┬─ASY──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               ├─SYN──────────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               ├─KEEPSRB──────────────────────────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               └─NKEEPSRB─────────────────────────────────────────────────────────────────────────────────────────────────────────
                                                                                                                                    
                                                                                                                                    
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.
5. KEEPSRB is meaningful only for synchronous operations.
6. NKEEPSRB is meaningful only for synchronous operations.
7. You can code more than one suboperand on OPTCD, but no more than one from each group.
8. See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-

any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

This field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

- X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG in the RPl extension.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

There are no state changes associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (REMOTE PROGRAM REPLIED AFFIRMATIVELY)
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE

RCPRI	RCSEC	Meaning
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ INVALID_FOR_FULL-DUPLEX_CONVERSATION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SEND, QUALIFY=CONFRMD

Purpose

This macroinstruction sends a positive confirmation reply to the remote application program on a half-duplex conversation.

Usage

This macroinstruction can only be used for half-duplex conversations.

When the application program receives a CONFIRM indication in the WHATRCV field after an APPCCMD CONTROL=RECEIVE macroinstruction, the application issues this macroinstruction to indicate that all the data that was sent by the CONFIRM indication has been received and is acceptable. This allows an application program to synchronize processing with its partner application.

If the application program receives a CONFIRM indication and it detects an error in the data it received before the CONFIRM, it can issue APPCCMD CONTROL=SEND, QUALIFY=ERROR to send a negative reply to the CONFIRM.

This macroinstruction corresponds to the CONFIRMED verb described in the LU 6.2 architecture.

Context

This macroinstruction may only be issued from the following conversation states on a half-duplex conversation:

- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE

This macroinstruction is not allowed on a full-duplex conversation.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
┌──────────┐
└─name─────┘──APPCCMD CONTROL=SEND──,QUALIFY=CONFRMD──────(2)──,RPL=──┬─rpl_address_field──────────┬──(8)──►
                                                                      └─(rpl_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   │                                                                                                          │
   │                                                                                                          │
   │                                                                                                          │
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                         │
   └─,CONMODE=──┬─BUFFCA──┬──────────────────────────────────────────────────────────────────────────────────────────────────────┘
                │─CS──────│
                │─LLCA────│
                └─SAME────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                         │
   └─,CONXMOD=──┬─CA──────┬──────────────────────────────────────────────────────────────────────────────────────────────────────┘
                │─CS──────│
                └─SAME────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                  (2)                                                                                         │
   └─,CONVID=──┬─32-bit_resource_id_field──────────────────────────────────────────────────────────────────────────────────────┘
               └─(32-bit_resource_id_register)──┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                              │
   │                                                                                                                              │
   │                                                                                                                              │
```

application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register*)
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program

immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*epl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of conversation. This field is labeled RPL6CCST in the RPL extension.

This field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

RECEIVE state is entered when an indicator of CONFIRM, DATA_CONFIRM, or DATA_COMPLETE_CONFIRM was received on the preceding APPCCMD CONTROL=RECEIVE.

SEND state is entered when an indicator of CONFIRM_SEND, DATA_CONFIRM_SEND, or DATA_COMPLETE_CONFIRM_SEND was received on the preceding APPCCMD CONTROL=RECEIVE.

END_CONV state is entered when an indicator of CONFIRM_DEALLOCATE, DATA_CONFIRM_DEALLOCATE, or DATA_COMPLETE_CONFIRM_DEALLOCATE was received on the preceding APPCCMD CONTROL=RECEIVE.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (PARTNER LU REPLIED AFFIRMATIVELY)
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE

RCPRI	RCSEC	Meaning
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ INVALID_FOR_FULL-DUPLEX_CONVERSATION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SEND, QUALIFY=DATA

Purpose

This macroinstruction sends data to a partner LU.

Usage

This macroinstruction transfers data that is specified by the AREA parameter into the SEND buffer of the conversation that is specified by the CONVID parameter. When there is more data in the conversation's SEND buffer than the maximum RU size for the conversation's session, an RU is sent to the partner LU. If the data does not exceed a maximum RU size, the data in the buffer remains there until the application program sends more data or causes the SEND buffer to be flushed.

Note: If OPTCD=XBUFLST is specified on this macroinstruction, all of the data is sent to the partner LU, even if the data does not exceed the maximum RU size.

The AREA parameter can specify a single data area to be sent, or it can specify a buffer list that points to multiple data areas to be sent. The OPTCD parameter specifies which of these methods is used.

For a complete discussion of sending data, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

This macroinstruction corresponds to the SEND_DATA verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND
- PENDING_SEND

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►                                                  (2)                          (8)
 ┌──────────┐APPCCMD CONTROL=SEND──,QUALIFY=DATA──────────,RPL=─┬─rpl_address_field──────────┬──►
 └─name─────┘                                                  └─(rpl_address_register)─────┘

►──┬──────────────
```

```
►──┬──────────────
```

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See "Coding Default Values" on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

When OPTCD=XBUFLST is specified on this macroinstruction, VTAM performs an internal flush of any data remaining in the send buffer, even if it does not exceed the maximum RU size.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource (that processes on the SEND/RECEIVE queue if the conversation is half-duplex or on the SEND queue if the conversation is full-duplex) until the command has completed. The exceptions to this are the APPCCMD CONTROL=REJECT, QUALIFY=CONV and the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstructions. The application can issue APPCCMDs against the same conversation resource that processes on the RECEIVE (if the conversation is full-duplex), EXPEDITED SEND, EXPEDITED RECEIVE, and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing the APPCCMD on a full-duplex conversation.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained

within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.
- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can contain the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV

The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

Note: The LOGRCV field is reserved if this macroinstruction is issued on a full-duplex conversation.

YES (B'1')

An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. The application program must perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data, and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')

Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI

The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC

The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV

A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD

The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE

The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA

The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field is labeled RPL6RSIG in the RPL extension. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')

A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by the remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_ CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE

RCPRI	RCSEC	Meaning
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00AC'	X'0001'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_PROGRAM
X'00AC'	X'0002'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_SERVICE
X'00AC'	X'0003'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_TIME
X'00AC'	X'0004'	ERROR_INDICATION_RECEIVED— ALLOCATION_ERROR
X'00AC'	X'0005'	ERROR_INDICATION_RECEIVED—UNKNOWN_ ERROR_CODE
X'00AC'	X'0006'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_RETRY
X'00AC'	X'0007'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_NO_RETRY
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED

RCPRI	RCSEC	Meaning
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=SEND, QUALIFY=DATACON

Purpose

This macroinstruction sends data that is supplied by the application program and any data that is already in the SEND buffer to a partner application program on a half-duplex conversation. The data is followed by a confirmation request.

Usage

This macroinstruction can only be used on a half-duplex conversation.

VTAM places the data specified by the AREA parameter in the SEND buffer of the conversation specified by the CONVID parameter. VTAM sends all data in the SEND buffer to the partner LU. The data is followed by a confirmation request. This macroinstruction completes only after a confirmation reply is received from the partner LU. The application program must ensure that the data that it sends completes a logical record.

For more information on sending and responding to confirmation requests, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

This macroinstruction corresponds to the SEND_DATA followed by CONFIRM verbs described in the LU 6.2 architecture.

Context

This macroinstruction can be issued on a half-duplex conversation from the following conversation states:

- SEND
- PENDING_SEND

This macroinstruction is not allowed on a full-duplex conversation.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
┬──────────┬──APPCCMD CONTROL=SEND──,QUALIFY=DATACON──────(2)──,RPL=┬─rpl_address_field──────────(8)─┬──►
└─name─────┘                                                        └─(rpl_address_register)─────────┘

►──┬──────────────
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │ ,AREA=──┬─data_area_or_buffer_list_address_field──────────────────────────────────────────────────────────────────────────────
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

  If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs

running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type

macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing the APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource that processes on the SEND/RECEIVE queue until the command has completed. The exception to this is the APPCCMD CONTROL=REJECT, QUALIFY=CONV macroinstruction. The application can issue APPCCMDs against the same conversation resource that processes on the EXPEDITED SEND, EXPEDITED RECEIVE and TESTSTAT queues. For more information about conversation queues refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each

entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.
- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of conversation. This field is labeled RPL6CCST in the RPL extension.

This field can contain the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be

received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data, and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING
X'0038'
PROGRAM_ERROR_TRUNC
X'003C'
SERVICE_ERROR_NO_TRUNC
X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA
The field in the RPL extension in which the signal code and signal extension

fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field is labeled RPL6RSIG in the RPL extension. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK (REMOTE PROGRAM REPLIED AFFIRMATIVELY)
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE

RCPRI	RCSEC	Meaning
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ INVALID_FOR_FULL-DUPLEX_CONVERSATION
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=SEND, QUALIFY=DATAFLU

Purpose

This macroinstruction sends data supplied by the application program as well as any data that is already in the SEND buffer to the partner application.

Usage

This macroinstruction combines the functions of two macroinstructions: APPCCMD CONTROL=SEND, QUALIFY=DATA followed by APPCCMD CONTROL=SEND, QUALIFY=FLUSH. VTAM places the data that is specified by the AREA parameter in the SEND buffer of the conversation that is specified by the CONVID parameter. VTAM sends all data in the SEND buffer to the partner LU.

This macroinstruction corresponds to SEND_DATA followed by FLUSH verbs described in the LU 6.2 architecture.

For a complete discussion of sending data, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND
- PENDING_SEND

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►                                                  (2)                                  (8)
 ┌──────────┐APPCCMD CONTROL=SEND──,QUALIFY=DATAFLU──────,RPL=┬─rpl_address_field──────┬►
 └─name─────┘                                                └─(rpl_address_register)─┘

►──┬──────────────
```

```
►──┬──────────────
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register***)**
Specifies the address of a data buffer or buffer list.

- If OPTCD=NBUFFLST, AREA specifies the address of an area containing the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=BUFFLST, AREA specifies the address of a buffer list. Each entry in the buffer list points to the data to be sent. Unless an HPDT request has preceeded this macroinstruction on this conversation, VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)
- If OPTCD=XBUFLST, AREA specifies the address of an extended buffer list. The data to be sent resides in CSM buffers. Once XBUFLST has been specified on an APPCCMD, VTAM does not track logical records supplied by the application on this or subsequent requests, for the duration of the conversation. Each entry in the extended buffer list is 48 bytes. RU boundaries and logical record boundaries are independent of the buffer boundaries. Each entry in the buffer list can specify any displacement in a CSM buffer. VTAM uses the CSM token rather than the storage address to track a given CSM buffer. Note that a CSM token cannot be repeated in an extended buffer list.

  If multiple areas of a CSM buffer are to be used on an APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction, which obtains additional tokens for the storage area. The tokens are provided on the extended buffer list and specified on the APPCCMD macroinstruction.

This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs

running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type

macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

CRYPT
Specifies whether data at the location indicated by the AREA is to be encrypted before it is sent on the conversation. This field is labeled RPLTCRYP in the RPL.

CRYPT=NO
Do not encrypt data before it is sent.

CRYPT=YES
Encrypt the data before it is sent. Specify CRYPT=YES only if encryption is allowed on the mode to which the conversation is allocated. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of how VTAM determines the level of cryptography.)

Note: If CRYPT=YES is specified, VTAM does not use HPDT services to transfer data, even if OPTCD=XBUFLST is specified. Instead, the normal send or receive path is used.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource (that processes on the SEND/RECEIVE queue if the conversation is half-duplex or on the SEND queue if the conversation is full-duplex) until the command has completed. The exceptions to this are the APPCCMD CONTROL=REJECT, QUALIFY=CONV and the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstructions. The application can issue APPCCMDs against the same conversation resource that processes on the RECEIVE (if the conversation is full-duplex), EXPEDITED SEND, EXPEDITED RECEIVE, and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing the APPCCMD on a full-duplex conversation.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST
Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

OPTCD=XBUFLST
Specifies that the HPDT interface is to be used. The AREA field of the RPL points to an extended buffer list containing 48-byte buffer list entries. Each entry in the buffer list points to a CSM buffer to be used for sending data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 48 bytes.

The following requirements apply to APPCCMD macroinstructions used to send data from an application-supplied extended buffer list:

- Applications using HPDT must use authorized path processing. Therefore, BRANCH=NO cannot be specified when OPTCD=XBUFLST.
- Entries in the extended buffer list must not contain any negative values. If a negative value exists in the entry, then the macroinstruction is rejected with an RCPRI, RCSEC combination of X'002C', X'0010' (INVALID DATA ADDRESS OR LENGTH).

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the data area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)
- If OPTCD=XBUFLST, RECLEN specifies the length of the extended buffer list that in turn points to the data to be sent. RECLEN must be a nonzero multiple of 48 bytes. (Extended buffer list entries consist of 48 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

Note: The LOGRCV field is reserved if this macroinstruction is issued on a full-duplex conversation.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. The application program must perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data, and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE

The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA

The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field is labeled RPL6RSIG in the RPL extension. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')

A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')

No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF

The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS

The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage

occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED

RCPRI	RCSEC	Meaning
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'002C'	PARAMETER_ERROR_INVALID_EXPEDITED_DATA_LENGTH
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0006'	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION
X'00AC'	X'0001'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_PROGRAM
X'00AC'	X'0002'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_SERVICE
X'00AC'	X'0003'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_TIME
X'00AC'	X'0004'	ERROR_INDICATION_RECEIVED— ALLOCATION_ERROR
X'00AC'	X'0005'	ERROR_INDICATION_RECEIVED—UNKNOWN_ ERROR_CODE
X'00AC'	X'0006'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_RETRY
X'00AC'	X'0007'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_NO_RETRY
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR—NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=SEND, QUALIFY=ERROR

Purpose

This macroinstruction informs the partner LU that the local application program detects an error.

Usage

When this macroinstruction is issued, VTAM builds an FMH-7, based on the TYPE and SENSE parameters, to represent the error that the application program detected.

The application program can specify one of the following types of errors:
- PROGRAM—error in an end-user transaction program
- SERVICE—error in a service component of a transaction program
- USER—user-specified error.

VTAM determines the sense code to place in the FMH-7 for program and service errors. The application program specifies the sense code on the SENSE parameter for user errors. The sense code specified must be appropriate to the error. Otherwise, improper processing of the macroinstruction might result. For a list of valid sense codes for an FMH-7, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

A negative response must be sent to the partner LU before the FMH-7 can be transmitted if the conversation is in one of the following states:
- RECEIVE
- PEND_SEND
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE

VTAM flushes the SEND buffer before the FMH-7 is created and a negative response is not sent if the conversation is in one of the following states:
- SEND
- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

For half-duplex conversations, the FMH-7 (and error log data that is supplied) is not sent to the partner LU until the application program issues a macroinstruction such as APPCCMD CONTROL=SEND, QUALIFY=FLUSH that causes the SEND buffer to be flushed. For full-duplex conversations, the FMH-7 is sent immediately to the conversation partner.

This macroinstruction corresponds to the SEND_ERROR verb described in the LU 6.2 architecture.

For more details on error handling, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND

- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PENDING_RECEIVE_LOG

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=SEND ,QUALIFY=ERROR (2) ,RPL= rpl_address_field (8)
                                                      (rpl_address_register)
►
 ,AARREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)
►
 ,ACB= acb_address_field (1)
       (acb_address_register)
►
 ,AREA= optional_log_data_area_address_field (1)
        (optional_log_data_area_address_register)
►
 ,BRANCH= NO (1)                ,CONMODE= BUFFCA (2)
          YES                            CS
                                         LLCA
                                         SAME
►
 ,CONVID= 32-bit_resource_id_field (2)      ,CONXMOD= CA (2)
          (32-bit_resource_id_register)             CS
                                                    SAME
►
 ,ECB= INTERNAL (3) (1)
       ecb_address_field
       (ecb_address_register)
 ,EXIT= exit_routine_address_field (4) (1)
        (exit_routine_address_register)
```

```
►──┬──────────────
```

associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*optional_log_data_area_address_field*

AREA=(*optional_log_data_area_address_register***)**
Specifies the address of a data area containing a formatted error log GDS variable to be sent to the partner LU. The application program is responsible for placing the error log data into the local system log. VTAM treats the error log GDS variable the same as other conversation data. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a description of the error log GDS variable.) This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|ANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|SPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|SPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|SPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|ANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record

format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource (that processes on the SEND/RECEIVE queue if the conversation is half-duplex or on the SEND queue if the conversation is full-duplex) until the command has completed. The exceptions to this are the APPCCMD CONTROL=REJECT, QUALIFY=CONV and the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstructions. The application can issue APPCCMDs against the same conversation resource that processes on the RECEIVE (if the conversation is full-duplex), EXPEDITED SEND, EXPEDITED RECEIVE, and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing the APPCCMD on a full-duplex conversation.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL. A 0 value in the RECLEN field indicates that the application program has chosen not to provide optional error log data to VTAM. If the application program specifies RECLEN=0, VTAM indicates in the FMH-7 it creates as a result of this APPCCMD that no error log data follows the FMH-7, and the AREA field in the RPL is ignored.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SENSE=*user-supplied_32-bit_fmh-7_sense_code*

SENSE=(*user-supplied_32-bit_fmh-7_sense_code_register***)**
Specifies the sense code that VTAM places in the FMH-7. This field is applicable only when TYPE=USER is specified. It is labeled RPL6SNSO in the RPL extension.

TYPE

Specifies the type of error being reported. This field is intended to distinguish between errors to be reported to end-user transaction programs and errors to be reported to a service component, such as a mapped conversation component, of the LU. This field is labeled RPL6TYPE in the RPL extension.

TYPE=PROGRAM

Specifies that an end-user transaction program error is being reported. VTAM determines the appropriate sense code to be placed in the FMH-7 based upon the state of the conversation and of the LU's SEND buffer. VTAM also determines whether the FMH-7 should be preceded by a negative response, based upon the current state of the conversation.

VTAM will place a sense code of either X'08890000' or X'08890001' in the FMH-7 for this type of error.

TYPE=SERVICE

Specifies that a service-component error is being reported. VTAM determines the appropriate sense code to be placed in the FMH-7 based upon the state of the conversation and of the LU's SEND buffer. VTAM also determines whether the FMH-7 should be preceded by a negative response, based upon the current state of the conversation.

VTAM will place a sense code of either X'08890100' or X'08890101' in the FMH-7 for this type of error.

TYPE=USER

Specifies that the application program is providing to VTAM a user-specified sense code that is to be placed in the FMH-7. The FMH-7 sense code is passed to VTAM through the SENSE field of the RPL extension. It is the responsibility of the application program to supply a valid FMH-7 sense code. This user-specified sense code must be appropriate for the error. Otherwise, improper processing of the macroinstruction might occur. VTAM determines whether the FMH-7 should be preceded by a negative response, based upon the current state of the conversation. For a list of sense codes that the application program can use, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

For more discussion on this type of error, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE

The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

Note: The LOGRCV field is reserved if this macroinstruction is issued on a full-duplex conversation.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. The application program must perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data, and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING

X'0044'
SERVICE_ERROR_TRUNC

X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. It is labeled RPL6SNSI in the RPL extension. This field has meaning only if the RCPRI field is set to a nonzero value. If the session for the conversation was deactivated, this code explains why. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. Not all RCPRI values have sense data associated with them. If the RCPRI field indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM.

SIGDATA
The field in the RPL extension in which the signal code and signal extension

fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

- Hex 00010001 indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV

The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field is labeled RPL6RSIG in the RPL extension. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')

A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')

No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF

The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS

The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied data buffer or buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD

Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

These changes are applicable when RCPRI indicates OK.

For half-duplex conversations, the conversation state is SEND after successful completion.

For full-duplex conversations, no conversation state changes occur.

See “Chapter 2. Return Codes” on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes. The return codes that can be returned depend on the state of the conversation at the time this APPCCMD is issued.

If APPCCMD CONTROL=SEND, QUALIFY=ERROR is issued in SEND state, the following values can be returned:

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	INVALID_LL
X'002C'	X'000B'	INCOMPLETE_GDS_VARIABLE_SUPPLIED
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0040'	X'0000'	SERVICE_ERROR_PURGING

RCPRI	RCSEC	Meaning
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_NEGATIVE_ RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

If APPCCMD CONTROL=SEND, QUALIFY=ERROR is issued in SEND/RECEIVE, SEND_ONLY, or PENDING_SEND/RECEIVE_log state, the following values can be returned:

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	INVALID_LL
X'002C'	X'000B'	INCOMPLETE_GDS_VARIABLE_SUPPLIED
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00AC'	X'0001'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_PROGRAM

RCPRI	RCSEC	Meaning
X'00AC'	X'0002'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_SERVICE
X'00AC'	X'0003'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_TIME
X'00AC'	X'0004'	ERROR_INDICATION_RECEIVED— ALLOCATION_ERROR
X'00AC'	X'0005'	ERROR_INDICATION_RECEIVED—UNKNOWN_ ERROR_CODE
X'00AC'	X'0006'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_RETRY
X'00AC'	X'0007'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_NO_RETRY

If APPCCMD CONTROL=SEND, QUALIFY=ERROR is issued in RECEIVE, PEND_SEND, or PEND_RCV_LOG state, the following values can be returned:

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	INVALID_LL
X'002C'	X'000B'	INCOMPLETE_GDS_VARIABLE_SUPPLIED
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0080'	X'0000'	DEALLOCATE_NORMAL
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

If APPCCMD CONTROL=SEND, QUALIFY=ERROR is issued in RECEIVE_CONFIRM, RECEIVE_CONFIRM_SEND, or RECEIVE_CONFIRM_DEALLOCATE state, the following values can be returned:

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	INVALID_LL
X'002C'	X'000B'	INCOMPLETE_GDS_VARIABLE_SUPPLIED

RCPRI	RCSEC	Meaning
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SEND, QUALIFY=FLUSH

Purpose

This macroinstruction flushes the VTAM SEND buffer associated with the specified conversation.

Usage

This macroinstruction is useful for optimizing processing between the application program and its partner LU. VTAM normally buffers the data from consecutive SEND macroinstructions until it has enough data for transmission. With this macroinstruction, the application program causes VTAM to transmit the data immediately.

VTAM flushes the buffer only when there is something in it. Issuing this macroinstruction when the SEND buffer is empty does not cause anything to flow to the partner LU.

For half-duplex conversations, VTAM buffers function management headers (FMH-5 and FMH-7). The FLUSH macroinstruction may be used to ensure that the headers are sent to the partner LU immediately.

Issuing an APPCCMD CONTROL=SEND, QUALIFY=FLUSH on a full-duplex conversation may cause the early completion of an APPCCMD CONTROL=RECEIVE, FILL=BUFF for the partner transaction program.

This macroinstruction corresponds to the FLUSH verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND
- PENDING_SEND

For full-duplex conversations, this macroinstruction can be issued from the following conversation states:
- SEND/RECEIVE
- SEND_ONLY
- PENDING_SEND/RECEIVE_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►                                                  (2)                                  (8)
┬──────┬──APPCCMD CONTROL=SEND──,QUALIFY=FLUSH──────,RPL=──┬─rpl_address_field──────────┬──►
└name  ┘                                                  └─(rpl_address_register)─────┘
```

```
►──┬──────────────
```

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE,

QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource (that processes on the SEND/RECEIVE queue if the conversation is half-duplex or on the SEND queue if the conversation is full-duplex) until the command has completed. The exceptions to this are the APPCCMD CONTROL=REJECT, QUALIFY=CONV and the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstructions. The application can issue APPCCMDs against the same conversation resource that processes on the RECEIVE (if the conversation is full-duplex), EXPEDITED SEND, EXPEDITED RECEIVE, and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing the APPCCMD on a full-duplex conversation.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register*)
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE

The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

FDB2

The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN

The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV

The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')

One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')

No FMH-5s are waiting to be received by the application program.

RCPRI

The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC

The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when

RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00AC'	X'0001'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_PROGRAM
X'00AC'	X'0002'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_SERVICE
X'00AC'	X'0003'	ERROR_INDICATION_RECEIVED—DEALLOCATE_ ABEND_TIME
X'00AC'	X'0004'	ERROR_INDICATION_RECEIVED— ALLOCATION_ERROR
X'00AC'	X'0005'	ERROR_INDICATION_RECEIVED—UNKNOWN_ ERROR_CODE
X'00AC'	X'0006'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_RETRY

RCPRI	RCSEC	Meaning
X'00AC'	X'0007'	ERROR_INDICATION_RECEIVED—RESOURCE_ FAILURE_NO_RETRY
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SEND, QUALIFY=RQSEND

Purpose

This macroinstruction notifies the partner LU on a half-duplex conversation that the local application program is requesting to enter SEND state. The conversation is changed to SEND state when the local application program subsequently receives a SEND indication from the remote application program.

Usage

This macroinstruction can only be used on half-duplex conversations.

When this macroinstruction is issued, VTAM sends a SIGNAL RU to the partner application program to indicate that the local LU is requesting to enter SEND state. The signal code and signal extension fields of the SIGNAL RU carry X'00010001'.

When the partner application receives the REQUEST_TO_SEND notification, it can enter RECEIVE state. When an APPCCMD CONTROL=RECEIVE macroinstruction completes and the SEND indicator is on in the WHATRCV field, the local application program is informed that it is in SEND state.

This macroinstruction can be issued while other macroinstructions are outstanding. However, if this macroinstruction is issued while an APPCCMD CONTROL=SEND, QUALIFY=RQSEND or an APPCCMD CONTROL=SENDEXPD macroinstruction is outstanding, it completes with return codes that indicate PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING.

This macroinstruction corresponds to the REQUEST_TO_SEND verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- SEND
- PENDING_SEND

This macroinstruction is not allowed on full-duplex conversations.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►                                                  (2)                          (8)
┬──────────┬──APPCCMD CONTROL=SEND──,QUALIFY=RQSEND──────,RPL=──┬─rpl_address_field──────────┬──►
└─name─────┘                                                   └─(rpl_address_register)─────┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,AAREA=──┬─rpl_extension_address_field──────────────────────────────────────────────────────────────────────────────────────
              │                                                                                                                    
              └─(rpl_extension_address_register)──┘                                                                               
                                                                                                                                   
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,ACB=──┬─acb_address_field──────────────────────────────────────────────────────────────────────────────────────────────────
            │                                                                                                                      
            └─(acb_address_register)──┘                                                                                            
                                                                                                                                   
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,BRANCH=──┬─NO──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────
               │     │                                                                                                              
               └─YES─┘                                                                                                              
                                                                                                                                   
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,CONMODE=──┬─BUFFCA──┬──────────────────────────────────────────────────────────────────────────────────────────────────────
                │         │                                                                                                          
                ├─CS────┤                                                                                                          
                │         │                                                                                                          
                ├─LLCA──┤                                                                                                          
                │         │                                                                                                          
                └─SAME──┘                                                                                                          
                                                                                                                                   
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,CONVID=──┬─32-bit_resource_id_field──────────────────────────────────────────────────────────────────────────────────────
               │                                                                                                                    
               └─(32-bit_resource_id_register)──┘                                                                                   
                                                                                                                                   
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   └─,CONXMOD=──┬─CA──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────
                │     │                                                                                                              
                ├─CS──┤                                                                                                              
                │     │                                                                                                              
                └─SAME┘                                                                                                              
                                                                                                                                   
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                                                                
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────
            │                                                                                                                      
            │                                                                                                                      
            │                                                                                                                      
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-

any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD

Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN

Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY

Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource that processes on the EXPEDITED SEND queue until the command has completed. The application can issue APPCCMDs against the same conversation resource that processes on the SEND/RECEIVE if the conversation is half-duplex, or the SEND and RECEIVE queues if the conversation is full-duplex, and the EXPEDITED RECEIVE and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing this APPCCMD.

OPTCD=KEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**

Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE

The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG

X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was

initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0009'	REQUEST_TERMINATED_BY_END_OF_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0004'	REQUEST_NOT_ALLOWED—CONTROL/QUALIFY_VALUE_ INVALID_FOR_FULL-DUPLEX_CONVERSATION
X'00A0'	X'0005'	REQUEST_NOT_ALLOWED—RSP_HAS_NOT_BEEN_ RECEIVED_FOR_A_PREVIOUS_SENDEXPD_REQUEST
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SENDEXPD, QUALIFY=DATA

Purpose

This macroinstruction sends expedited data to a partner LU over a full-duplex or a half-duplex conversation established on a *full-duplex-capable session*. If the session is not full-duplex capable, an RCPRI, RCSEC combination of X'00A0', X'0001', REQUEST_NOT_ALLOWED—LU_PAIR_DOES_NOT_SUPPORT_ SENDING_EXPEDITED_DATA is returned to the application.

Usage

The amount of expedited data specified by the application should be in the range of 1–86 bytes. If the length of the expedited data is outside of this range, an RCRPI, RCSEC combination of X'002C', X'002C', PARAMETER_ERROR—INVALID_EXPEDITED_DATA_LENGTH is returned to the application.

This macroinstruction will be posted complete immediately without waiting for a response from the partner LU. A response will not be sent by the partner until the expedited data has been received by the partner application.

If the conversation ends before the macroinstruction has a chance to process, RCRPI, RCSEC combination of X'0000', X'0009', REQUEST_TERMINATED_BY_END_OF_CONVERSATION is returned to the application.

If this macroinstruction is issued while another APPCCMD CONTROL=SENDEXPD macroinstruction or an APPCCMD CONTROL=SEND, QUALIFY=RQSEND macroinstruction is currently outstanding for the specified conversation, an RCRPI, RCSEC combination of X'002C', X'0011', PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING is returned to the application.

If the EXPEDITED SEND queue has been prohibited, then an RCRPI, RCSEC combination of X'00A0', X'0002', REQUEST_NOT_ALLOWED—REQUEST_BLOCKED, is returned to the application.

An RCRPI, RCSEC combination of X'0050', X'0000', STATE_ERROR, will be returned when the macroinstruction is issued in PENDING_DEALLOCATE state.

If the macroinstruction is issued and the response to a previously issued SENDEXPD request has not been received, then an RCPRI, RCSEC combination of X'00A0', X'0005', REQUEST_NOT_ALLOWED—RSP_HAS_NOT_BEEN_RECEIVED_ FOR_A_PREVIOUS_SENDEXPD_REQUEST is returned to the application.

This macroinstruction will always cause a flow.

This macroinstruction corresponds to the SEND_EXPEDITED_DATA verb described in the LU 6.2 architecture.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- SEND

- RECEIVE
- RECEIVE_CONFIRM
- RECEIVE_CONFIRM_SEND
- RECEIVE_CONFIRM_DEALLOCATE
- PENDING_END_CONV_LOG
- PENDING_SEND
- PENDING_RECEIVE_LOG

For full-duplex conversation, this macroinstruction can be issued form the following conversations states:
- SEND/RECEIVE
- SEND_ONLY
- RECEIVE_ONLY
- PENDING_SEND/RECEIVE_LOG
- PENDING_RECEIVE-ONLY_LOG
- PENDING_RESET_LOG

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 ┌──────────┐
 └─name─────┘──APPCCMD CONTROL=SENDEXPD──,QUALIFY=DATA──────(2)──,RPL=──┬─rpl_address_field──────────┬──(8)──►
                                                                        └─(rpl_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   └─,AAREA=──┬─rpl_extension_address_field──────────────────────────────────────────────────────────────(1)─┘
              └─(rpl_extension_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   └─,ACB=──┬─acb_address_field──────────────────────────────────────────────────────────────────────────(1)─┘
            └─(acb_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   └─,AREA=──┬─data_area_or_buffer_list_address_field──────────────────────────────────────────────────────(1)─┘
             └─(data_area_or_buffer_list_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   └─,BRANCH=──┬─NO──┬──────────────────────────────────────────────────────────────────────────────────(1)─┘
               └─YES─┘
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────┐
   └─,CONMODE=──┬─BUFFCA─┬──────────────────────────────────────────────────────────────────────────────(2)─┘
                ├─CS─────┤
                ├─LLCA───┤
                └─SAME───┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────┬──►
   └─,CONVID=──┬─32-bit_resource_id_field──────────────────────────────────────────────────────────────(2)─┘
               └─(32-bit_resource_id_register)─┘
   ┌──────────────────────────────────────────────────────────────────────────────────────────────────────┐
   └─,CONXMOD=──┬─CA─┬──────────────────────────────────────────────────────────────────────────────────(2)─┘
                ├─CS─┤
                └─SAME─┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                   (3)    (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────────────────────────────────────────────────────────────────────────────────────────
   │        └─(ecb_address_register)──────────────────────────────────────────────────────────────────────────────────────────
   │                                                                                   (4)    (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)──────────────────────────────────────────────────────────────────────────────────

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                  ┌──────────────────────────────────────────────────────────────────────────────────────────────────────┐
   └─,OPTCD=(──────────┴─,──────────────────────────────────────────────────────────────────────────────────────────────────────┴─)──┬
                       │   (7)
                       │        (1)
                       ├─ASY──────────────────────────────────────────────────────────────────────────────────────────────────────
                       │   (1)
                       │ │
                       │ └─SYN──────────────────────────────────────────────────────────────────────────────────────────────────
                       │        (1)
                       ├─BUFFLST──────────────────────────────────────────────────────────────────────────────────────────────────
                       │   (1)
                       │ │
                       │ └─NBUFFLST─────────────────────────────────────────────────────────────────────────────────────────────
                       │        (1)   (5)
                       ├─KEEPSRB──────────────────────────────────────────────────────────────────────────────────────────────────
                       │   (1)   (6)
                       │ │
                       │ └─NKEEPSRB─────────────────────────────────────────────────────────────────────────────────────────────

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                   (1)
   └─,RECLEN=──┬─data_length──────────────────────────────────────────────────────────────────────────────────────────────────────
               └─(data_length_register)──────────────────────────────────────────────────────────────────────────────────────────
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register*)
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_or_buffer_list_address_field*

AREA=(*data_area_or_buffer_list_address_register*)
Specifies the address of a data buffer or buffer list. If OPTCD=NBUFFLST, AREA specifies the address of a data area containing the data to be sent. If OPTCD=BUFFLST, AREA specifies the address of a buffer list that in turn points to the data to be sent. This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and

that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=SAME
Specifies that the conversation mode for expedited information is to remain unchanged at the completion of this macroinstruction.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD

Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN

Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY

Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource that processes on the EXPEDITED SEND queue until the command has completed. The application can issue APPCCMDs against the same conversation resource that processes on the SEND/RECEIVE if the conversation is half-duplex, or the SEND and RECEIVE queues if the conversation is full-duplex, and the EXPEDITED RECEIVE and TESTSTAT queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing this APPCCMD.

OPTCD=KEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB

Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST

Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. The indicator resides within the RPLOPT6 field of the RPL.

If OPTCD=BUFFLST is chosen, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The RECLEN field of the RPL specifies a buffer list length that is a nonzero multiple of 16 bytes. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries.

OPTCD=NBUFFLST

Specifies that the data supplied by the application program is contained within a single buffer area. The AREA field specifies the address of the buffer and the RECLEN field specifies the length of the buffer. The indicator resides within the RPLOPT6 field of the RPL.

RECLEN=*data_length*

RECLEN=(*data_length_register***)**
Specifies the length of the data to be sent or the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=NBUFFLST, RECLEN specifies the number of bytes of data to be sent from the buffer area specified by AREA.
- If OPTCD=BUFFLST, RECLEN specifies the length of the buffer list that in turn points to the data to be sent. The RECLEN specifies a buffer list length that is a nonzero multiple of 16 bytes. (Buffer list entries consist of 16 bytes.)

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES so long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field is labeled RPL6RSIG in the RPL extension. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture. The indication is either YES or NO (RPL6RSIG bit set on or off).

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by the remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0009'	REQUEST_TERMINATED_BY_END_OF_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'002C'	PARAMETER_ERROR—INVALID_EXPEDITED_DATA_LENGTH
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0001'	REQUEST_NOT_ALLOWED—LU_PAIR_DOES_NOT_ SUPPORT_SENDING_EXPEDITED_DATA
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A0'	X'0005'	REQUEST_NOT_ALLOWED—RSP_TO_PREVIOUS_ REQUEST_NOT_RECEIVED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SENDFMH5, QUALIFY=NULL

Purpose

This macroinstruction accepts and sends an FMH-5 for a conversation reserved by the APPCCMD CONTROL=PREALLOC macroinstruction.

Usage

This macroinstruction completes the allocation of a conversation begun by a previous APPCCMD CONTROL=PREALLOC. VTAM does not activate any additional session between the application program and it's partner LU as a result of this command.

The APPCCMD CONTROL=SENDFMH5 macroinstruction does not return any vectors to the application in the vector area. For conversations on half-duplex-capable sessions, the FMH-5 is stored in the SEND buffer. For conversations on full-duplex-capable sessions, the FMH-5 is flushed immediately.

Context

This macroinstruction can only be issued from the PENDING_ALLOCATE conversation state.

Syntax

```
►►                                                  (2)
 [name] APPCCMD CONTROL=SENDFMH5,QUALIFY=NULL ──┬──────────────────────────────────────────►
                                                │                                (8)
                                                └,RPL=──┬─rpl_address_field──────┬──┘
                                                        └─(rpl_address_register)─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └,AAREA=──┬─rpl_extension_address_field──────────┬──┘
             └─(rpl_extension_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   │                                  (1)
   └,ACB=──┬─acb_address_field──────────┬──┘
           └─(acb_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   │                                                                              (1)
   └,AREA=──┬─fmh-5_and_optional_pip_gds_variable_address_field──────────┬──┘
            └─(fmh-5_and_optional_pip_gds_variable_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   │                  (1)                                                         (2)
   └,BRANCH=──┬─NO──┬──┘   └,CONVID=──┬─32-bit_resource_id_field──────────┬──┘
              └─YES─┘                 └─(32-bit_resource_id_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   │                                  (3)   (1)
   ├,ECB=──┬─INTERNAL──────────────────┬──┘
   │       ├─ecb_address_field─────────┤
   │       └─(ecb_address_register)────┘
   │                                                  (4)   (1)
   └,EXIT=──┬─exit_routine_address_field──────────┬──┘
            └─(exit_routine_address_register)─────┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                   ┌──,──────────────────────────────┐
   │                                   ▼                                 │
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────┴──────────────────────────────────────────────────────────)─┘
               │                   (7)                                   │
               │                   (1)                                   │
               ├──ASY──────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
               │                   (1)                                   │
               ├──SYN──────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
               │                   (1)   (5)                             │
               ├──KEEPSRB──────────────────────────────────────────────────────────────────────────────────────────────────────────┤
               │                   (1)   (6)                             │
               └──NKEEPSRB─────────────────────────────────────────────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────
```

AREA specifies the address of a data area containing the data to be sent. If OPTCD=BUFFLST, AREA specifies the address of a buffer list that in turn points to the data to be sent. In either case, the data consists of logical records. VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.) This field is labeled RPLAREA in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RECLEN=*fmh-5_and_optional_gds_field_length*

RECLEN=(*fmh-5_and_optional_gds_field_length_register***)**
Specifies the length of the data within the data area indicated by the AREA field. This field is labeled RPLRLEN in the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of the RPL and RPL extension fields:

CONSTATE
The field in the RPL extension that indicates what state the conversation is in. It is labeled RPL6CCST in the RPL extension.

This field can have the following values for half-duplex conversations:
X'00' RESET
X'01' SEND
X'08' END_CONVERSATION

This field can have the following values for full-duplex conversations:
X'00' RESET
X'80' FDX_RESET
X'81' SEND/RECEIVE

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SLS
The field in the RPL extension that indicates whether the session was established using session-level LU-LU verification. This field is labeled RPL6SLS in the RPL extension.

YES (B'1')
The session was established using session-level LU-LU verification.

NO (B'0')
The session was not established using session-level LU-LU verification.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

State Changes

After successful completion of this macroinstruction, the conversation state is SEND if issued over a half-duplex session or SEND/RECEIVE if issued over a full-duplex session.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0006'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_ SUPPORTED_BY_LU
X'0004'	X'0000'	ALLOCATION_ERROR—ALLOCATION_FAILURE_NO_ RETRY
X'0004'	X'0010'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_ VALID_FOR_FULL-DUPLEX
X'0004'	X'0011'	ALLOCATION_ERROR—LU_PAIR_NOT_ SUPPORTING_FDX_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID CONVERSATION
X'002C'	X'000A'	PARAMETER_ERROR—INCOMPLETE_FMH5_SUPPLIED
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_ OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0015'	PARAMETER_ERROR—INVALID_TPN
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0022'	PARAMETER_ERROR—INVALID_CONTROL_OR_ QUALIFY_VALUE
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0048'	X'0000'	RESOURCE_FAILURE,_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE,_RETRY
X'0050'	X'0000'	STATE_ERROR
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0074'	X'0000'	HALT_ISSUED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE.

APPCCMD CONTROL=SENDRCV, QUALIFY=DATAFLU

Purpose

This macroinstruction provides a dual function; it performs the function of a send, and when the send is complete it automatically performs the function of a receive.

The send portion of this macroinstruction sends data supplied by the application program and any data that is already in the SEND buffer to the partner application. After the send portion of this macroinstruction is successfully completed, the conversation is placed in receive state and the macroinstruction waits for data from the partner.

This macroinstruction can only be issued for half-duplex conversations.

Usage

This macroinstruction combines the functions of two macroinstructions: APPCCMD CONTROL=SEND, QUALIFY=DATAFLU followed by APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC. A buffer list format must be used to allow the application program to specify areas and lengths separately for both the send and receive portions of this macroinstruction. For a description of how to use both non-extended buffer list entries and extended buffer list entries refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

When this macroinstruction is issued, VTAM places data in the SEND buffer of the conversation that is specified by the CONVID parameter. VTAM determines the location of the data to be sent from the buffer list entries specified by the AREA parameter. VTAM sends all data in the SEND buffer to the partner LU.

When the send portion of this macroinstruction completes, there is no data ready to be received on the conversation; therefore, VTAM queues the macroinstruction until data arrives. This macroinstruction has just turned the flow around and the SEND indication is still enroute to the partner. After the partner receives the data just sent and also the SEND indication, it may then send data back to the local application. When enough of this data is received by VTAM to satisfy the receive portion of this macroinstruction the macroinstruction will be completed.

After data is received, VTAM copies any received data from the conversation that is specified by the CONVID parameter to the data area that is specified by the last entry in the buffer list.

When this macroinstruction completes, the BLERECLN field of the last buffer list entry indicates how much data was written to the data area. The WHATRCV field indicates what type of data was received.

The application program can issue this macroinstruction when the conversation is in SEND or PENDING_SEND state. VTAM flushes its SEND buffer, sending all buffered information, along with the SEND indicator, to the partner LU. This changes the conversation to RECEIVE state. VTAM then waits for information to arrive. The remote application program can send data to the local application program after it receives the SEND indication.

For a complete discussion of sending data and receiving data, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

For half-duplex conversations, this macroinstruction can be issued from the following conversation states:

- SEND
- PENDING_SEND

For full-duplex conversations, this macroinstruction is not allowed.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=SENDRCV,QUALIFY=DATAFLU (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
► [,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1)] ►
► [,ACB= [acb_address_field | (acb_address_register)] (1)] ►
► [,AREA= [buffer_list_address_field | (buffer_list_address_register)] (1)] [,BRANCH= [NO | YES] (1)] ►
► [,CD= [DEFER | IMMED] (2)] [,CONMODE= [BUFFCA | CS | LLCA | SAME] (2)] ►
► [,CONVID= [32-bit_resource_id_field | (32-bit_resource_id_register)] (2)] [,CONXMOD= [CA | CS] (2)] ►
► [,ECB= [INTERNAL | ecb_address_field | (ecb_address_register)] (3) (1)
   [,EXIT= [exit_routine_address_field | (exit_routine_address_register)] (4) (1)] [,FILL= [BUFF | LL] (2)] ►
```

```
►─,OPTCD=(──┬─BUFFLST─┬──┬──────────────────────────────────┬──)──────────────────────►
            └─XBUFLST─┘  │ (7)                              │
                         │ ┌──────────────────────────────┐ │
                         │ │      (1)                     │ │
                         │ │ ┌─ASY──────────────────────┐ │ │
                         │ │ │      (1)                 │ │ │
                         │ │ └─SYN──────────────────────┘ │ │
                         │ │            (1)  (5)          │ │
                         │ │ ┌─KEEPSRB──────────────────┐ │ │
                         │ │ │            (1)  (6)      │ │ │
                         │ │ └─NKEEPSRB─────────────────┘ │ │
                         └─┴──────────────────────────────┴─┘

►──┬──────────────────────────────────────────────────────────────────────────────────►◄
   │                                   (1)
   └─,RECLEN=──┬─data_length──────────────────┬──
               └─(data_length_register)───────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 Refer to "Coding Default Values" in *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=*rpl_extension_address_register*
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=*acb_address_register*
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*buffer_list_address_field*

AREA=*buffer_list_address_register*

Specifies the address of a list of buffer entries.

- If **OPTCD=BUFFLST**, all entries in the buffer list except the last specify the address and length of data to be sent. The data consists of logical records. VTAM tracks the logical records supplied by the application program, examining the logical-record length (LL) field associated with each logical record. (It does not inspect the data portion of the logical record.)

  The last entry specifies the address and length of an area in which data is to be received. When this macroinstruction completes, another field in this last entry contains the number of bytes placed in this receive buffer by VTAM.

  Both the send and receive buffers are described using the ISTBLENT DSECT. For a more detailed description of how to use buffer list entries refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.
- If **OPTCD=XBUFLST**, all entries in the buffer list except the last specify the address and length of data to be sent. The send data resides in CSM buffers. VTAM does not track logical records supplied by the application.

  Like OPTCD=BUFFLST, the last entry specifies the address and length of an area in which data is to be received. When this macroinstruction completes, another field in this last entry contains the number of bytes placed in this receive buffer by VTAM. This receive buffer is not a CSM buffer.

  The send buffers are described using the ISTBLXEN DSECT and the receive buffer is described using the ISTBLENT DSECT. For a more detailed description of how to use extended buffer list entries, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

When the application program receives information other than data, as indicated by the WHATRCV field of the RPL extension, nothing is placed in this data area. This field is labeled RPLAREA in the RPL.

BRANCH

Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO

Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES

Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CD

This parameter controls subsequent actions if a SEND indication is received in the WHATRCV field on the receive portion of this macroinstruction. For this to happen, the send portion of this macroinstruction transmitted the SEND indication to the partner, as is normally done on this macroinstruction, which in turn returned it. The SEND indication is being reported back to the local application on the receive portion of this macroinstruction. In particular CD specifies whether the LU immediately goes to SEND or whether the LU defers the SEND transition by going into PEND_SEND when a change of direction is received with no data.

CD=DEFER
Specifies that the conversation state will be in PEND_SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CD=IMMED
Specifies that the conversation state will be in SEND state when the SEND indicator of the WHATRCV field is set and none of the data indicators are set.

CONMODE
Specifies the mode for receiving normal information upon completion of the APPCCMD. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that only APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC can be used to receive data on this conversation. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY can be used to receive data on this conversation and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC macroinstruction.

CONMODE=SAME
Specifies that the continuation mode of the conversation is to remain unchanged.

CONVID=*32-bit_resource_id_field*

CONVID=*32-bit_resource_id_register*
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD. This field is labeled RPL6CXMD in the RPL extension.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received by either a specific-type macroinstruction or an any-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC|ISPEC or APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY|IANY.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can be received only by a specific-type macroinstruction, such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=*exit_routine_address_register*
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

FILL
Specifies whether the application program is to receive data in terms of the logical-record format of the data. This parameter applies only to the receive portion of this macroinstruction and corresponds to FILL=LL|BUFFER described in the LU 6.2 architecture. This field is labeled RPL6FILL in the RPL extension.

FILL=BUFF
Specifies the application program is to receive data independently of its logical-record format. FILL=BUFF corresponds to FILL=BUFFER on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

FILL=LL
Specifies the application program is to receive one logical record, or whatever portion of the logical record is available. FILL=LL corresponds to FILL=LL on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the

scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

In general, when the application program regains control after issuing an asynchronous APPCCMD, it is prevented from issuing another APPCCMD against the same conversation resource until the prior asynchronous command has completed. The exceptions to this are the APPCCMD CONTROL=SEND, QUALIFY=RQSEND; APPCCMD CONTROL=REJECT; and the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstructions. (For more information, refer to the descriptions of the particular macroinstructions). The application program is allowed to issue APPCCMDs against other conversations.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=BUFFLST
Specifies that the data supplied by the application program is contained within multiple buffers. This option allows the application program to provide data from discontiguous buffer areas. RU boundaries are independent of the buffer boundaries. VTAM creates RUs based upon the maximum SEND RU size regardless of whether the data is taken from one buffer, part of a buffer, or multiple buffers. Logical records are also independent of the buffer boundaries. This field is labeled RPLBUFFL in the RPL. When OPTCD=BUFFLST, the AREA field of the RPL points to a buffer list that is a contiguous set of 16-byte control blocks, called buffer list entries. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a detailed description of these buffer list entries.) The buffer list created by the application must have at least two entries. One or more entries must be send buffer list entries. This specifies the layout of the send buffers. The last entry must be a special receive entry that points to the receive buffer and indicates the area length. Both the send buffer(s) and the receive buffer are described by the ISTBLENT macroinstruction. The following explains the layout of the receive entry:

- The first 4 bytes are reserved and should be set to 0 when the macroinstruction is issued. This field will be used to return the amount of data received to the application.
- The second 4 bytes contain the length of the receive buffer. This is similar to the AREALEN field of an RPL that accompanies a receive type macroinstruction.
- The third 4 bytes contain the address of a receive buffer. This is similar to the AREA field that accompanies a receive type macroinstruction.
- The fourth 4 bytes must contain zeros (the send length field).

OPTCD=XBUFLST
Specifies that the data supplied by the application program is contained within an extended buffer list. The AREA field of the RPL points to an extended buffer list that contains a contiguous set of 48-byte send extended buffer list entries followed immediately by a 16-byte receive buffer entry. Once OPTCD=XBUFLST has been issued, VTAM no longer tracks logical records for the duration of the conversation.

The indicator is labeled RPLXBFL and resides within the RPLOPT6 field of the RPL.

Each send entry in the extended buffer list can point to any displacement into a CSM buffer and is described by ISTBLXEN. VTAM uses the CSM token rather than the storage address to track a CSM buffer. A CSM token cannot be repeated in an extended buffer list. If multiple areas of a CSM buffer are to be used on one APPCCMD, the CSM buffer must first be segmented by using the IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction. This macroinstruction returns a new token for each CSM buffer segment. The new tokens should then be used on the APPCCMD. VTAM treats the CSM storage associated with the new CSM tokens as separate CSM buffers.

The last entry describes the receive buffer. This buffer is not a CSM buffer. It is described using the ISTBLENT DSECT.

RECLEN
Specifies the length of the buffer list containing the data to be sent. This field is labeled RPLRLEN in the RPL.

- If OPTCD=BUFFLST, the length of the buffer list is determined by the product of 16 and the number of entries, both send and receive. (Each buffer list entry consists of 16 bytes.)
- If OPTCD=XBUFLST, the length of the buffer list is determined by the product of 48 and the number of send entries plus 16 bytes for the receive buffer entry. (Each CSM buffer list entry consists of 48 bytes.)

RPL=*rpl_address_field*

RPL=*rpl_address_register*
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

The following conversation states are possible:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE CONFIRM
X'04' RECEIVE CONFIRM_SEND
X'05' RECEIVE CONFIRM_DEALLOC
X'07' PENDING_END_CONVERSATION_LOG
X'08' END CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

EXPDLEN
The field in the RPL6 that shows the length of the expedited data waiting to be received. This field has meaning only when EXPDRCV=YES. This field is labeled RPL6EXDL in the RPL extension.

EXPDRCV
The field in the RPL6 that indicates whether expedited data is waiting to be received. This field is labeled RPL6EXDR in the RPL extension.

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner application programs. The FMH5RCV field continues to be set to YES as long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

LOGRCV
The field in the RPL extension that returns an indication of whether error log data is expected. The indication is either YES or NO (RPL6RLOG set on or off). This field is labeled RPL6RLOG in the RPL extension.

YES (B'1')
An FMH-7 was received that specified that error log data follows. The application program must issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC in order to retrieve the log data. It is the responsibility of the application program to perform an optional receive check after issuing APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to determine whether the expected log data was sent by the partner LU. The data must be error log data and it must be in the form of a GDS variable.

LOGRCV=YES only if the RCPRI field of the RPL extension contains one of the following values:

X'0004'
ALLOCATION_ERROR

X'0014'
DEALLOCATE_ABEND_PROGRAM

X'0018'
DEALLOCATE_ABEND_SERVICE

X'001C'
DEALLOCATE_ABEND_TIMER

X'0030'
PROGRAM_ERROR_NO_TRUNC

X'0034'
PROGRAM_ERROR_PURGING

X'0038'
PROGRAM_ERROR_TRUNC

X'003C'
SERVICE_ERROR_NO_TRUNC

X'0040'
SERVICE_ERROR_PURGING
X'0044'
SERVICE_ERROR_TRUNC
X'0048'
RESOURCE_FAILURE,_NO_RETRY
X'005C'
USER_ERROR_CODE_RECEIVED

NO (B'0')
Either no error indicator was received or an error indicator was received but indicated that no log data follows.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field used on the SEND portion of this macroinstruction, similar to a send with a buffer list. It is used to calculate the number of entries in the buffer list.

For the receive portion of this macroinstruction, VTAM calculates a RECLEN value but does not overlay the RECLEN provided by the application in the RPL. Instead, VTAM returns the receive RECLEN in the first 4 bytes of the last entry in the buffer list (BFERECLN), which is the entry used to describe the receive area.

RPLXSRV
A field in the RPL that is set if VTAM accepts all the CSM buffers from the application on an HPDT request. If the APPCCMD completes unsuccessfully and the completion status is stored in the RPL, the application must examine RPLXSRV. Some TPEND exits are driven where the RPL is canceled and not posted complete. It is the application's responsibility to examine the RPLXSRV bit and determine if CSM storage needs to be freed.

For more information about application recovery options when RPLXSRV is not set, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

The RPLXSRV indicator is contained in the RPLEXTDS field in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

SENSE
The field in the RPL extension that returns a 32-bit sense code. This field has meaning only if the RCPRI field is set to a nonzero value. The sense code also can be set when the return code is RESOURCE_FAILURE_NO_RETRY. This code indicates why the session for the conversation was deactivated. Not all RCPRI values have sense data associated with them. If the RCPRI field

indicates USER_ERROR_CODE_RECEIVED, the SENSE field contains an FMH-7 sense code that was not interpreted by VTAM. It is labeled RPL6SNSI in the RPL extension.

SIGDATA
The field in the RPL extension in which the signal code and signal extension fields of a received SIGNAL RU are returned to the application program. This field has meaning only when SIGRCV=YES. This field is labeled RPL6SGNL in the RPL extension.

X'00010001' indicates a REQUEST_TO_SEND notification has been received from the remote application program.

Note: The SIGDATA field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

SIGRCV
The field in the RPL extension that returns an indication of whether an application program's partner has requested permission to send. This field and the SIGDATA field correspond to the REQUEST_TO_SEND_RECEIVED parameter described in the LU 6.2 architecture.

Note: The SIGRCV field is reserved if, on the macroinstruction that initiated the conversation (APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5), the application specified RTSRTRN=EXPD.

The indication is either YES or NO (RPL6RSIG bit set on or off). It is labeled RPL6RSIG.

YES (B'1')
A SIGNAL RU has been received from the partner LU. The values carried in the signal code and signal extension fields of the SIGNAL RU are returned to the application program in the SIGDATA field.

NO (B'0')
No SIGNAL RU has been received from the partner LU. When SIGRCV=NO, the SIGDATA field contains no meaningful information.

STSHBF
The field in the RPL extension that returns the address of the current buffer. It is used with STSHDS to give the current position (address and displacement) in the application-supplied buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STBF in the RPL extension.

STSHDS
The field in the RPL extension that returns the displacement into the current buffer. It is used with STSHBF to give the current position (address and displacement) in the application-supplied buffer list (the area pointed to by the AREA field of the RPL) when a temporary storage shortage occurs while data is being sent. All data prior to this buffer has been sent. This field is labeled RPL6STDS in the RPL extension.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the

conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by a remote application program). This field is labeled RPL6USR in the RPL extension.

Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.

WHATRCV

The field in the RPL extension that returns a mask specifying what the application program received. It is labeled RPL6WHAT. The application program should examine this WHATRCV mask only when RCPRI indicates X'0000'. Otherwise, WHATRCV has no meaning.

When RCPRI indicates OK, one or more bits in the mask can be turned *on* (contain a value of B'1') to indicate the type of information that has been received. For instance, if the application program is being passed both conversation data and a request for confirmation, both the DATA and CONFIRM bits will be set *on*; the other bits will be set *off*.

The 2-byte WHATRCV mask has the following format.

RPL6RCV1

Bit	Meaning
0	DATA
1	DATA_COMPLETE
2	DATA_INCOMPLETE
3	SEND
4	CONFIRM
5	DEALLOCATE
6	LOG_DATA
7	PS_HEADER

RPL6RCV2

Bit	Meaning
0	PARTIAL_PS_HEADER
1–7	Reserved

For example, a WHATRCV value indicating that DATA has been received would be represented by X'8000'. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for a discussion of the meaning of this field.

State Changes

See the description of the WHATRCV mask for state changes when RCPRI indicates OK.

See "Chapter 2. Return Codes" on page 585 for state changes associated with other return codes.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See "Chapter 2. Return Codes" on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0004'	X'0002'	ALLOCATION_ERROR—CONVERSATION_TYPE_ MISMATCH
X'0004'	X'0003'	ALLOCATION_ERROR—PIP_NOT_ALLOWED
X'0004'	X'0004'	ALLOCATION_ERROR—PIP_NOT_SPECIFIED_CORRECTLY
X'0004'	X'0005'	ALLOCATION_ERROR—SECURITY_NOT_VALID

RCPRI	RCSEC	Meaning
X'0004'	X'0007'	ALLOCATION_ERROR—SYNC_LEVEL_NOT_SUPPORTED_ BY_PROGRAM
X'0004'	X'0008'	ALLOCATION_ERROR—TPN_NOT_RECOGNIZED
X'0004'	X'0009'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_NO_ RETRY
X'0004'	X'000A'	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
X'0004'	X'000B'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_NO_RETRY
X'0004'	X'000C'	ALLOCATION_ERROR—CANNOT_RECONNECT_TRANS_ PGM_RETRY
X'0004'	X'000D'	ALLOCATION_ERROR—RECONNECT_NOT_SUPPORTED_ BY_PGM
X'0014'	X'0000'	DEALLOCATE_ABEND_PROGRAM
X'0018'	X'0000'	DEALLOCATE_ABEND_SERVICE
X'001C'	X'0000'	DEALLOCATE_ABEND_TIMER
X'0024'	X'0000'	LOGICAL_RECORD_BOUNDARY_ERROR
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0003'	PARAMETER_ERROR—INVALID_LL
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_ LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'0012'	PARAMETER_ERROR—BUFFER_LIST_LENGTH_INVALID
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0024'	PARAMETER_ERROR—PS_HEADER_NOT_SUPPLIED
X'002C'	X'0025'	PARAMETER_ERROR—PS_HEADER_LENGTH_IS_ INSUFFICIENT
X'002C'	X'0028'	PARAMETER_ERROR—CRYPTOGRAPHY_NOT_ALLOWED_ ON_MODE
X'002C'	X'0031'	PARAMETER_ERROR—SENDRCV_SPECIFIED_WITHOUT_ OPTCD=BUFFLST|XBUFLST
X'002C'	X'0032'	PARAMETER_ERROR— UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0030'	X'0000'	PROGRAM_ERROR_NO_TRUNC
X'0034'	X'0000'	PROGRAM_ERROR_PURGING
X'0038'	X'0000'	PROGRAM_ERROR_TRUNC
X'003C'	X'0000'	SERVICE_ERROR_NO_TRUNC
X'0040'	X'0000'	SERVICE_ERROR_PURGING
X'0044'	X'0000'	SERVICE_ERROR_TRUNC
X'0048'	X'0000'	RESOURCE_FAILURE_NO_RETRY
X'004C'	X'0000'	RESOURCE_FAILURE_RETRY
X'0050'	X'0000'	STATE_ERROR
X'005C'	X'0000'	USER_ERROR_CODE_RECEIVED—FOLLOWING_ NEGATIVE_RESPONSE
X'005C'	X'0001'	USER_ERROR_CODE_RECEIVED—WITHOUT_NEGATIVE_ RESPONSE
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE_OR_RESOURCE_ SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOCATE_ABEND

RCPRI	RCSEC	Meaning
X'008C'	X'0000'	PARTNER_COMMITTED_PROTOCOL_VIOLATION
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'0094'	X'0000'	INVALID_CONDITION_FOR_SENDING_DATA
X'0098'	X'0000'	STORAGE_SHORTAGE_WHILE_SENDING_DATA
X'00A0'	X'0004'	CONTROL/QUALIFY_VALUE_INVALID_FOR_ FULL-DUPLEX_CONVERSATION
X'00A0'	X'0006'	REQUEST_NOT_ALLOWED—PROGRAM_NOT_AUTHORIZED_ FOR_REQUESTED_FUNCTION
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE
X'00B4'	X'0001'	CSM_DETECTED_ERROR— NOT_SPECIFIED
X'00B4'	X'0002'	CSM_DETECTED_ERROR— INVALID_BUFFER_TOKEN_SPECIFIED
X'00B4'	X'0003'	CSM_DETECTED_ERROR— INVALID_INSTANCE_ID_SPECIFIED

APPCCMD CONTROL=SETSESS, QUALIFY=RESUME

Purpose

This macroinstruction resumes sending any outgoing normal data that was held because APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND was issued previously on the specified session.

Usage

This macroinstruction should be issued to notify VTAM to allow any outbound normal data to flow to the partner if any has been held due to a previously issued APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND command. APPCCMD CONTROL=SETSESS, QUALIFY=RESUME also enables the following to resume:

- Normal data flow from any conversations matched to the session
- Normal session deactivation
- Session bidding

This macroinstruction indicates to VTAM that the application program (which is supporting a sync point manager) has completed its synchronization processing successfully.

APPCCMD CONTROL=REJECT, QUALIFY=SESSION can be issued if the application program's synchronization processing was unsuccessful and the application program does not wish to imply by the normal data flow that the sync point completed successfully.

If this macroinstruction is issued and the session has not been suspended, a return code of 0 is received, but no changes are made.

Context

This macroinstruction is not conversation-specific and therefore is not conversation-state-driven.

Syntax

```
►►
  [name] APPCCMD CONTROL=SETSESS ,QUALIFY=RESUME (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
►
  [,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1)] ►
►
  [,ACB= [acb_address_field | (acb_address_register)] (1)] [,BRANCH= [NO | YES] (1)] ►
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                   (3)   (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────────┤
   │        └─(ecb_address_register)─────────┘
   │                                                                                   (4)   (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)──────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                  ┌──────────────────────────────────────────────────────────────────────────────────────────┐
   │                  │ ,                                                                                        │
   │                  │   (7)                                                                                    │
   └─,OPTCD=(─────────┴──┬──────────────────────────────────────────────────────────────────────────────────────┴─)┘
                         │          (1)
                         ├─┬─ASY──────────────────────────────────────────────────────────────────────────────────
                         │ │          (1)
                         │ └─SYN──────────────────────────────────────────────────────────────────────────────────┘
                         │                 (1)   (5)
                         ├─┬─KEEPSRB──────────────────────────────────────────────────────────────────────────────
                         │ │                 (1)   (6)
                         │ └─NKEEPSRB─────────────────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                                                   (2)
   └─,SESSID=──┬─session_instance_id_field────────────────────────────────────────────────────────────────────────
               └─(session_instance_id_register)─────────────────────────────────────────────────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                                                                   (2)
   └─,SESSIDL=──┬─session_instance_id_length────────────────────────────────────────────────────────────────────────
                └─(session_instance_id_length_register)──────────────────────────────────────────────────────────┘
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register*)
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register*)
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SESSID=*session_instance_id_field*

SESSID=(*session_instance_id_register***)**
Specifies the session to which this macroinstruction applies. The session instance identifier, which was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction, indicates the session to be released. This field is labeled RPL6SSID in the RPL extension.

SESSIDL=*session_instance_id_length*

SESSIDL=(*session_instance_id_length_register***)**
Specifies the length of the session instance ID. The value specified must be greater than 0 and less than or equal to 8. The session instance ID length was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction. This field is labeled RPL6SIDL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0023'	PARAMETER_ERROR—INVALID_SESSION_INSTANCE_ IDENTIFIER
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND

Purpose

This macroinstruction specifies that the application program wants VTAM to suspend any outgoing normal data flow on the specified session after the current conversation has been deallocated. APPCCMD CONTROL=SETSESS, QUALIFY=RESUME resumes the outgoing normal flow.

Usage

This macroinstruction should be issued to notify VTAM to not allow outbound flow on the session. It should be issued if the application program (which is supporting a sync point manager) has not completed the synchronization processing needed before the partner can continue its synchronization processing. The application program must issue this command before the conversation supporting the sync point exchange is deallocated to ensure the flow is stopped on the free session.

Suspending the session gives the application program with the sync point manager control of the outbound flow whose subsequent receipt at the partner implies a successful sync point has completed. The partner application program can then continue synchronization cleanup. Further information on the sync point services function is described in the *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*.

APPCCMD CONTROL=SETSESS, QUALIFY=RESUME indicates that the application program is ready to resume normal flow because its sync point processing completed successfully. APPCCMD CONTROL=REJECT, QUALIFY=SESSION can be issued if the sync point processing is unsuccessful.

If an application program is executing under persistent LU-LU session support and the application program fails after APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND has been issued and APPCCMD CONTROL=SETSESS, QUALIFY=RESUME has not been issued, VTAM UNBINDs the session and deallocates the conversation on which the synchronization is taking place. In the same situation, VTAM also UNBINDs sync point sessions for which APPCCMD CONTROL=SETSESS, QUALIFY=SYNCBEG has been issued but neither APPCCMD CONTROL=SETSESS, QUALIFY=SYNCEND nor APPCCMD CONTROL=SETSESS, QUALIFY=RESUME has been issued at the time of the failure.

Context

This macroinstruction is not conversation-specific and, therefore, is not driven by the conversation state.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 [name] APPCCMD CONTROL=SETSESS ,QUALIFY=SUSPEND (2) ,RPL= [rpl_address_field (8)
                                                           (rpl_address_register)]
```

```
►──┬──────────────
```

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an

asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SESSID=*session_instance_id_field*

SESSID=(*session_instance_id_register***)**
Specifies the session to which this macroinstruction applies. The session instance identifier, which was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction, indicates the session to be held. This field is labeled RPL6SSID in the RPL extension.

SESSIDL=*session_instance_id_length*

SESSIDL=(*session_instance_id_length_register***)**
Specifies the length of the session instance ID. The value specified must be greater than 0 and less than or equal to 8. The session instance ID length was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction. This field is labeled RPL6SIDL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

State Changes

There are no state changes associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0023'	PARAMETER_ERROR—INVALID_SESSION_INSTANCE_ IDENTIFIER
X'002C'	X'0026'	PARAMETER_ERROR—SESSION_INSTANCE_IDENTIFIER_ AND_CONVERSATION_ID_ARE_MISMATCHED
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0088'	X'0000'	CANCELLED_BY_REJECT_OR_DEALLOC_ABEND
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SETSESS, QUALIFY=SYNCBEG

Purpose

This macroinstruction notifies VTAM that a sync point exchange is beginning.

If an application program is executing under persistent LU-LU session support, persistence must be overridden for a session during the time that a sync point exchange takes place. If the application program fails after APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND has been issued and APPCCMD CONTROL=SETSESS, QUALIFY=RESUME has not been issued, VTAM UNBINDs the session and deallocates the conversation on which the synchronization is taking place. In the same situation, VTAM also UNBINDs sync point sessions for which APPCCMD CONTROL=SETSESS, QUALIFY=SYNCBEG has been issued, but neither APPCCMD CONTROL=SETSESS, QUALIFY=SYNCEND nor APPCCMD CONTROL=SETSESS, QUALIFY=RESUME has been issued at the time of the failure.

Usage

This macroinstruction is issued to notify VTAM that the sync point manager is beginning a synchronization exchange because a SYNCPT is being issued or a TAKE-SYNCPT is being received. To ensure that synchronization protocols are followed, VTAM UNBINDs this session when the application program fails, even though the application program has enabled persistence. The UNBIND permits the LUs to make consistent decisions and ensures continued synchronization between the two LUs. If the data is critical enough to use a synchronization exchange, APPCCMD CONTROL=SETSESS, QUALIFY=SYNCBEG and APPCCMD CONTROL=SETSESS, QUALIFY=SYNCEND should be used. For circumstances for use, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction is not conversation-specific and, therefore, is not driven by the conversation state. It performs a useful function only for application programs that are using persistent LU-LU sessions. If application programs that have not enabled persistence issue this macroinstruction, a good return code is sent but no action is taken.

This macroinstruction is not allowed for conversations pending deallocation for persistent LU-LU sessions.

Syntax

```
►►
 ┌──────┐
 └name  ┘─APPCCMD CONTROL=SETSESS──,QUALIFY=SYNCBEG──────(2)──,RPL=─┬─rpl_address_field──────────(8)─┬──►
                                                                    └─(rpl_address_register)─────────┘

►──┬──────────────
```

```
►──┬──────────────
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SESSID=*session_instance_id_field*

SESSID=(*session_instance_id_register***)**
Specifies the session to which this macroinstruction applies. The session instance identifier, which was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction, indicates the session to be released. This field is labeled RPL6SSID in the RPL extension.

SESSIDL=*session_instance_id_length*

SESSIDL=(*session_instance_id_length_register***)**
Specifies the length of the session instance ID. The value specified must be greater than 0 and less than or equal to 8. The session instance ID length was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction. This field is labeled RPL6SIDL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0002'	INVALID_CONVERSATION
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0023'	PARAMETER_ERROR—INVALID_SESSION_INSTANCE_ IDENTIFIER
X'002C'	X'0026'	SESSION_INSTANCE_IDENTIFIER_AND_CONVERSATION_ IDENTIFIER_MISMATCH
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0084'	X'0000'	STORAGE_SHORTAGE
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=SETSESS, QUALIFY=SYNCEND

Purpose

This macroinstruction indicates to VTAM that the sync point exchange has completed.

Note: This macroinstruction only has meaning for MVS™ and VSE applications using persistent sessions. VTAM ignores this macroinstruction if issued from a VM application.

Usage

This macroinstruction is issued to notify VTAM that the sync point exchange has completed, whether successful or not, and that VTAM no longer needs to UNBIND sync point sessions during a failure after persistence has been enabled. It is used with APPCCMD CONTROL=SETSESS, QUALIFY=SYNCBEG. For circumstances for use, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Context

This macroinstruction is not conversation-specific and, therefore, is not driven by conversation state. It performs a useful function only for application programs that are using persistent LU-LU sessions. If application programs that have not enabled persistence issue this macroinstruction, a good return code is sent but no action is taken.

Syntax

```
►►
 [name] APPCCMD CONTROL=SETSESS ,QUALIFY=SYNCEND (2) ,RPL= rpl_address_field (8) ►
                                                          (rpl_address_register)
►
 ,AAREA= rpl_extension_address_field (1)
         (rpl_extension_address_register)
►
 ,ACB= acb_address_field (1)                ,BRANCH= NO (1)
       (acb_address_register)                       YES
►
 ,ECB= INTERNAL (3) (1)
       ecb_address_field
       (ecb_address_register)
 ,EXIT= exit_routine_address_field (4) (1)
        (exit_routine_address_register)
```

```
►──┬──────────────
```

programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the

completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

SESSID=*session_instance_id_field*

SESSID=(*session_instance_id_register***)**
Specifies the session to which this macroinstruction applies. The session instance identifier, which was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction, indicates the session to be released. This field is labeled RPL6SSID in the RPL extension.

SESSIDL=*session_instance_id_length*

SESSIDL=(*session_instance_id_length_register***)**
Specifies the length of the session instance ID. The value specified must be greater than 0 and less than or equal to 8. The session instance ID length was passed to the application program on a previous APPCCMD CONTROL=ALLOC or APPCCMD CONTROL=RCVFMH5 macroinstruction. This field is labeled RPL6SIDL in the RPL extension.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_ OUTSTANDING
X'002C'	X'001F'	PARAMETER_ERROR—APPCCMD_ISSUED_FOR_ NON-APPC
X'002C'	X'0023'	PARAMETER_ERROR—INVALID_SESSION_INSTANCE_ IDENTIFIER
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=TESTSTAT, QUALIFY=ALL

Purpose

This macroinstruction obtains status on information from any active conversation. VTAM will wait for information to arrive on a conversation to satisfy the macroinstruction request. If information is available to be received, the application will receive status on the information without waiting.

Usage

The information returned from this macroinstruction is contained in the status data structure control block, CITY-STATE. The address of the control block must be specified in the RPLAREA field which can be set with the AREA keyword. See “Status Data Structure (ISTSTATD)” on page 648 for a description of the control block.

If the length of the area specified by the application is not sufficient to receive the entire status data structure (AREALEN should be a least 48 bytes) an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT is returned to the application. RECLEN will contain the length of the data structure.

Upon successful completion, this macroinstruction will return status on one or more of the following types of information:

- Normal information
- Expedited information (data and/or Request_To_Send Received)

If this macroinstruction is issued while another TESTSTAT ALL|IALL is currently outstanding, an RCPRI, RCSEC combination of X'002C', X'0011', PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING is returned to the application program.

This macroinstruction will not alter the conversation.

Context

Input states are not applicable to this macroinstruction.

Syntax

```
►►
 [name] APPCCMD CONTROL=TESTSTAT ,QUALIFY=ALL(2) ,RPL= rpl_address_field(8)
                                                       [(rpl_address_register)]
►──────────────────────────────────────────────────────────────────────────►
 [,AAREA= rpl_extension_address_field(1)
          [(rpl_extension_address_register)]]
►──────────────────────────────────────────────────────────────────────────►
 [,ACB= acb_address_field(1)
        [(acb_address_register)]]
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                   (1)
   └─,AREA=──┬─data_area_address_field──────────────────┬──────────────────────────────────────────────────────────────────────
             └─(data_area_address register)──────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                   (1)                                   (1)
   └─,AREALEN=──┬─data_area_length──────────────────────┬──────────────────────────────────────────────────────────────────────
                └─(data_area_length_register)──────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                   (3)    (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────────────────────────────────────────────────────────────────────────────────────────
   │        └─(ecb_address register)──────────────────────────────────────────────────────────────────────────────────────────
   │                                                                                   (4)    (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────────────────────────────────────────────────────
             └─(exit_routine_address register)──────────────────────────────────────────────────────────────────────────────────

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                  ┌──────────────────────────────────────────────────────────────────────────────────────────────────────┐
   │                  ▼                                                                                                      │
   └─,OPTCD=(──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────┬─)──┘
               │                                   (7)
               │                                                                                                              │
               │                                   (1)
               ├─┬─ASY──────────────────────────────────────────────────────────────────────────────────────────────────────┤
               │ │                                 (1)
               │ └─SYN──────────────────────────────────────────────────────────────────────────────────────────────────────┤
               │                                                 (1)    (5)
               ├─┬─KEEPSRB──────────────────────────────────────────────────────────────────────────────────────────────────┤
               │ │                                                 (1)    (6)
               │ └─NKEEPSRB─────────────────────────────────────────────────────────────────────────────────────────────────┤
```

Notes:

1 Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.

2 Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.

3 ECB is meaningful only for asynchronous operations.

4 EXIT is meaningful only for asynchronous operations.

5 KEEPSRB is meaningful only for synchronous operations.

6 NKEEPSRB is meaningful only for synchronous operations.

7 You can code more than one suboperand on OPTCD, but no more than one from each group.

8 See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. The data returned should be mapped using the status data structure, CITY-STATE. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. The application program must receive at least 48 bytes of data, or it will be rejected. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES so long as an FMH-5 is waiting to be

received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual size of the structure containing the status information VTAM placed in the AREA if the RCPRI,RCSEC fields equal X'0000', X'0000'. If the RCPRI,RCSEC fields equal X'002C', X'0008' RECLEN indicates the length of the status data structure, but because the receive buffer is not sufficient to contain the entire structure, none of the status data structure is returned to the application program. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD.

State Changes

No state changes are associated with this macroinstruction.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING
X'002C'	X'002E'	PARAMETER_ERROR—VECTOR_AREA_NOT_VALID
X'002C'	X'002F'	PARAMETER_ERROR—VECTOR_AREA_LENGTH_INSUFFICIENT
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE

RCPRI	RCSEC	Meaning
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=TESTSTAT, Qualifiable

Purpose

This macroinstruction obtains status on information immediately available from any active conversation. VTAM will not wait for information to arrive on a conversation to satisfy the macroinstruction request.

Usage

The information returned from this macroinstruction is contained in the status data structure control block, CITY-STATE. The address of the control block must be specified in the RPLAREA field, which can be set with the AREA keyword. See “Status Data Structure (ISTSTATD)” on page 648 for a description of the control block.

If the length of the area specified by the application is not sufficient to receive the entire status data structure (AREALEN should be a least 48 bytes), an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT is returned to the application. RECLEN will contain the length of the data structure.

If this macroinstruction is issued and information is not available on any conversation, an RCPRI, RCSEC combination of X'0000', X'0008', NO_IMMEDIATELY_AVAILABLE_INFORMATION is returned to the application.

Upon successful completion, this macroinstruction will return status on one or more of the following types of information:

- Normal information
- Expedited information (data and/or Request_To_Send Received)

If this macroinstruction is issued while another TESTSTAT ALL|IALL is currently outstanding, an RCPRI, RCSEC combination of X'002C', X'0011', PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING is returned to the application program.

This macroinstruction will not alter the conversation.

Context

Input states are not applicable to this macroinstruction.

Syntax

```
►►
 [name] APPCCMD CONTROL=TESTSTAT ,QUALIFY=IALL (2) ,RPL= rpl_address_field (8)
                                                        (rpl_address_register)
►
 [,AAREA= rpl_extension_address_field (1)
          (rpl_extension_address_register)]
```

```
►──┬──────────────────────────────────────────────────────────────────────────────►
   │                                   (1)
   └─,ACB=──┬─acb_address_field──────────┬──────────────────────────────────────
            └─(acb_address_register)─────┘

►──┬──────────────────────────────────────────────────────────────────────────────►
   │                                           (1)
   └─,AREA=──┬─data_area_address_field──────────┬──────────────────────────────
             └─(data_area_address_register)──────┘

►──┬──────────────────────────────────────────────────────────────────────────────►
   │                                                   (1)
   └─,AREALEN=──┬─data_area_length──────────────────────┬──────────────────────────
                └─(data_area_length_register)────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────►
   │                                   (3)    (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────┬──────────────────────────────────────
   │        └─(ecb_address_register)──────┘
   │                                                          (4)    (1)
   └─,EXIT=──┬─exit_routine_address_field──────────────────────────────────────
             └─(exit_routine_address_register)──────────────────────────────────

►──┬──────────────────────────────────────────────────────────────────────────────►◄
   │                  ┌─,──────────────────────┐
   │                  │   (7)                  │
   └─,OPTCD=(─────────┴────────────────────────┴─)──┬──────────────────────────
                      │          (1)                │
                      ├─ASY──────────────────────────┤
                      │   ┌──(1)──┐                 │
                      │   └─SYN───┘                 │
                      │              (1)   (5)      │
                      ├─KEEPSRB──────────────────────┤
                      │              (1)   (6)      │
                      └─NKEEPSRB─────────────────────┘
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.
5. KEEPSRB is meaningful only for synchronous operations.
6. NKEEPSRB is meaningful only for synchronous operations.
7. You can code more than one suboperand on OPTCD, but no more than one from each group.
8. See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register*)
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register*)
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register*)
Specifies the data area in which the application program is to receive the data. The data returned should be mapped using the status data structure, CITY-STATE. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register*)
Specifies the length value that is the maximum amount of data the application program is to receive. The application program must receive at least 48 bytes of data, or it will be rejected. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register*)
Specifies that VTAM is to post an event control block (ECB) when an

asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5

has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES so long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual size of the structure containing the status information VTAM placed in the AREA if the RCPRI,RCSEC fields equal X'0000', X'0000'. If the RCPRI,RCSEC fields equal X'002C', X'0008' RECLEN indicates the length of the status data structure, but because the receive buffer is not sufficient to contain the entire structure, none of the status data structure is returned to the application program. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. It is labeled RPLRTNCD.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0008'	NO_IMMEDIATELY_AVAILABLE_INFORMATION
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED

RCPRI	RCSEC	Meaning
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=TESTSTAT, QUALIFY=ISPEC

Purpose

This macroinstruction obtains status on information immediately available on a specified conversation. VTAM will not wait for information to arrive to satisfy the macroinstruction request.

Usage

The information returned from this macroinstruction is contained in the status data structure control block, CITY-STATE. The address of the control block must be specified in the RPLAREA field which can be set with the AREA keyword. See “Status Data Structure (ISTSTATD)” on page 648 for a description of the control block.

If the length of the area specified by the application is not sufficient to receive the entire status data structure (AREALEN should be a least 48 bytes) an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT is returned to the application. RECLEN will contain the length of the data structure.

If information is not available, an RCPRI, RCSEC combination of X'0000', X'0008', NO_IMMEDIATELY_AVAILABLE_INFORMATION is returned to the application program.

If the conversation ends before this macroinstruction can query the information received, if any, an RCPRI, RCSEC combination of X'0000', X'0009', REQUEST_TERMINATED_BY_END_OF_CONVERSATION is returned to the application.

Upon successful completion, this macroinstruction will return status on one or more of the following types of information:

- Normal information
- Expedited information (data and/or Request_To_Send Received)

This macroinstruction will not alter the conversation.

Context

This macroinstruction can be issued in any conversation state while the conversation is active so long as another APPCCMD CONTROL=TESTSTAT, QUALIFY=SPEC|ISPEC macroinstruction is not currently outstanding for the specified conversation.

Syntax

```
►►
 [name] APPCCMD CONTROL=TESTSTAT ,QUALIFY=ISPEC (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8)
►
 [,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1)]
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,ACB=──┬─acb_address_field──────────────────────┬──────────────────────────────────────────────────────────────────
            └─(acb_address_register)─────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,AREA=──┬─data_area_address_field────────────────┬──────────────────────────────────────────────────────────────────
             └─(data_area_address_register)───────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,AREALEN=──┬─data_area_length───────────────────┬──────────────────────────────────────────────────────────────────
                └─(data_area_length_register)────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,BRANCH=──┬─NO──┬──────────────────────────────────────────────────────────────────────────────────────────────────
               └─YES─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (2)
   └─,CONVID=──┬─32-bit_resource_id_field────────────┬──────────────────────────────────────────────────────────────────
               └─(32-bit_resource_id_register)──────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (3)   (1)
   │ ,ECB=──┬─INTERNAL──────────────────────────────┬──────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field─────────────────────┤
   │        └─(ecb_address_register)────────────────┘
   │                                                  (4)   (1)
   └─,EXIT=──┬─exit_routine_address_figure──────────┬──────────────────────────────────────────────────────────────────
             └─(exit_routine_address_register)──────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                                  ┌─,──────────────────────────┐
   │                                  │   (7)                      │
   └─,OPTCD=(─────────────────────────┴────────────────────────────┴─)──┬──────────────────────────────────────────────┬
                                        │                                │
                                        │      (1)                       │
                                        ├──ASY──────────────────────────┤
                                        │      (1)                       │
                                        │  ┌──┬──────────────────────┐  │
                                        │  └─SYN──────────────────────┘  │
                                        │            (1)   (5)           │
                                        ├──KEEPSRB──────────────────────┤
                                        │            (1)   (6)           │
                                        │  ┌──────────────────────────┐  │
                                        │  └─NKEEPSRB─────────────────┘  │
```

Notes:

1. Operand value might be placed in its RPL field either by specification on an RPL macroinstruction operand or by explicitly setting the field using the IFGRPL DSECT.
2. Operand value might be placed in its RPL extension field either by specification on an ISTRPL6 macroinstruction operand or by explicitly setting the field using the ISTRPL6X DSECT.
3. ECB is meaningful only for asynchronous operations.
4. EXIT is meaningful only for asynchronous operations.
5. KEEPSRB is meaningful only for synchronous operations.
6. NKEEPSRB is meaningful only for synchronous operations.
7. You can code more than one suboperand on OPTCD, but no more than one from each group.
8. See “Coding Default Values” on page 3 for information on coding operands on the RPL or the APPCCMD macroinstruction.

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. The data returned should be mapped using the status data structure, CITY-STATE. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. The application program must receive at least 48 bytes of data, or it will be rejected. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot

specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource that processes on the TESTSTAT queue until the command has completed. The application can issue APPCCMDs against the same conversation resource that processes on the SEND/RECEIVE queue if the conversation is half-duplex, or the SEND and RECEIVE queues if the conversation is full-duplex, and the EXPEDITED RECEIVE and EXPEDITED SEND queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing this APPCCMD.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension. For half-duplex conversations, this field can have the following values:

X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can contain the following values:

X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES so long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual size of the structure containing the status information VTAM placed in the AREA if the RCPRI,RCSEC fields equal X'0000', X'0000'. If the RCPRI,RCSEC fields equal X'002C', X'0008' RECLEN indicates the length of the status data structure, but because the receive buffer is not sufficient to contain the entire structure, none of the status data structure is returned to the application program. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by the remote application program). This field is labeled RPL6USR in the RPL extension.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0008'	NO_IMMEDIATELY_AVAILABLE_INFORMATION
X'0000'	X'0009'	REQUEST_TERMINATED_BY_END_OF_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING
X'002C'	X'0032'	PARAMETER_ERROR—UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD

RCPRI	RCSEC	Meaning
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

APPCCMD CONTROL=TESTSTAT, QUALIFY=SPEC

Purpose

This macroinstruction obtains status on information available on a specified conversation. VTAM will wait for information to arrive to satisfy the macroinstruction request. If information is already available, the application program receives status on it without waiting.

Usage

The information returned from this macroinstruction is contained in the status data structure control block, CITY-STATE. The address of the control block must be specified in the RPLAREA field, which can be set with the AREA keyword. See “Status Data Structure (ISTSTATD)” on page 648 for a description of the control block.

If the length of the area specified by the application is not sufficient to receive the entire status data structure (AREALEN should be a least 48 bytes), an RCPRI, RCSEC combination of X'002C', X'0008', PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT is returned to the application. RECLEN will contain the length of the data structure.

Upon successful completion, this macroinstruction will return status on one or more of the following types of information:

- Normal information
- Expedited information (data and/or Request_To_Send Received)

If the conversation is terminated before information is received, an RCPRI, RCSEC combination of X'0000', X'0009', REQUEST_TERMINATED_BY_END_OF_CONVERSATION is returned to the application program.

This macroinstruction will not alter the conversation.

Context

This macroinstruction can be issued in any conversation state while the conversation is active so long as another APPCCMD CONTROL=TESTSTAT QUALIFY=SPEC|ISPEC macroinstruction is not currently outstanding for the specified conversation.

Syntax

```
►►
 [name] APPCCMD CONTROL=TESTSTAT ,QUALIFY=SPEC (2) ,RPL= [rpl_address_field | (rpl_address_register)] (8) ►
► [ ,AAREA= [rpl_extension_address_field | (rpl_extension_address_register)] (1) ] ►
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,ACB=──┬─acb_address_field──────────────────────┬──────────────────────────────────────────────────────────
            └─(acb_address_register)─────────────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,AREA=──┬─data_area_address_field────────────────┬──────────────────────────────────────────────────────────
             └─(data_area_address_register)───────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,AREALEN=──┬─data_area_length───────────────────┬──────────────────────────────────────────────────────────
                └─(data_area_length_register)────────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (1)
   └─,BRANCH=──┬─NO──┬──────────────────────────────────────────────────────────────────────────────────────────
               └─YES─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                                  (2)
   └─,CONVID=──┬─32-bit_resource_id_field────────────┬──────────────────────────────────────────────────────────
               └─(32-bit_resource_id_register)──────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │                                   (3)    (1)
   ├─,ECB=──┬─INTERNAL──────────────────────────────────────────────────────────────────────────────────────────
   │        ├─ecb_address_field──────────────────┤
   │        └─(ecb_address_register)──────────────┘
   │                                                                  (4)    (1)
   └─,EXIT=──┬─exit_routine_address_field──────────┬──────────────────────────────────────────────────────────
             └─(exit_routine_address_register)────┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
   │                  ┌─,──────────────────────┐
   │                  │   (7)                  │
   └─,OPTCD=(─────────┴────────────────────────┴─────────────────────────────────────────────────────────────)──┘
                      │        (1)
                      ├─ASY──┬──────────────────────────────────────────────────────────────────────────────────
                      │      │        (1)
                      │      └─SYN────┬──────────────────────────────────────────────────────────────────────
                      │               │
                      │               │
                      │               │
```

Input Parameters

Following are descriptions of the input parameters:

AAREA=*rpl_extension_address_field*

AAREA=(*rpl_extension_address_register***)**
Specifies the address of the LU 6.2 RPL extension that will be associated with this APPCCMD macroinstruction. This field is labeled RPLAAREA in the RPL.

ACB=*acb_address_field*

ACB=(*acb_address_register***)**
Specifies the address of an access method control block that identifies the application program that is issuing the APPCCMD macroinstruction. VTAM associates conversations with application programs using the conversation ID (CONVID). The application program associates conversations with transaction programs. Application programs cannot issue APPCCMD macroinstructions in address spaces other than the ACB address space. This field is labeled RPLDACB in the RPL.

AREA=*data_area_address_field*

AREA=(*data_area_address_register***)**
Specifies the data area in which the application program is to receive the data. The data returned should be mapped using the status data structure, ISTSTATD. This field is labeled RPLAREA in the RPL.

AREALEN=*data_area_length*

AREALEN=(*data_area_length_register***)**
Specifies the length value that is the maximum amount of data the application program is to receive. The application program must receive at least 48 bytes of data, or it will be rejected. This field is labeled RPLBUFL in the RPL.

BRANCH
Specifies whether authorized path processing is to be used for application programs running in supervisor state under a TCB. Application programs running in TCB-mode supervisor state can use BRANCH=YES to obtain authorized path services. The indicator resides within the RPLEXTDS field of the RPL.

BRANCH=NO
Authorized path processing is not to be used. For application programs running in problem state (non-supervisor state) under a TCB, BRANCH=NO is the only option.

BRANCH=YES
Authorized path processing is to be used. For application programs running under an SRB rather than under a TCB, the macroinstruction is processed in this manner automatically, regardless of the actual setting of the BRANCH field.

CONVID=*32-bit_resource_id_field*

CONVID=(*32-bit_resource_id_register***)**
Specifies the resource ID of the conversation. This field is labeled RPL6CNVD in the RPL extension.

ECB
Valid only if OPTCD=ASY. Specifies how the application program requests to be informed of the completion of the APPCCMD macroinstruction. You cannot

specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLOPT1 field of the RPL.

ECB=INTERNAL
Specifies that VTAM is to post an internal ECB when the APPPCCMD macroinstruction completes.

ECB=*ecb_address_field*

ECB=(*ecb_address_register***)**
Specifies that VTAM is to post an event control block (ECB) when an asynchronous APPCCMD completes. *Event_control_block_address* is the location of the ECB to be posted. The ECB can be any fullword of storage aligned on a fullword boundary.

EXIT=*exit_routine_address_field*

EXIT=(*exit_routine_address_register***)**
Valid only if OPTCD=ASY. It indicates the address of a routine to be scheduled when the APPCCMD completes. You cannot specify both ECB and EXIT on a single APPCCMD macroinstruction. The indicator resides within the RPLEXTDS field of the RPL.

OPTCD
Specifies the following processing options that can be selected for the macroinstruction request:

OPTCD=SYN
Specifies that control is to be returned synchronously to the application program when the function of the APPCCMD has completed. The indicator resides within the RPLOPT1 field of the RPL.

OPTCD=ASY
Specifies that control is to be returned to the application program immediately and that the application program is to be informed later of the completion of the macroinstruction by the posting of an ECB or the scheduling of an exit. The indicator resides within the RPLOPT1 field of the RPL.

When the application program regains control after issuing an APPCCMD asynchronously, it is prevented from issuing another APPCCMD against the same conversation resource that processes on the TESTSTAT queue until the command has completed. The application can issue APPCCMDs against the same conversation resource that processes on the SEND/RECEIVE queue if the conversation is half-duplex, or the SEND and RECEIVE queues if the conversation is full-duplex, and the EXPEDITED RECEIVE and EXPEDITED SEND queues. For more information about conversation queues, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*

The application program is allowed to issue APPCCMDs against other conversations. OPTCD=ASY is recommended when issuing this APPCCMD.

OPTCD=KEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM returns to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

OPTCD=NKEEPSRB
Specifies that for a synchronous request issued in SRB mode, VTAM does not return to the application under the same SRB in which VTAM was invoked. The indicator resides within the RPLOPT11 field of the RPL.

RPL=*rpl_address_field*

RPL=(*rpl_address_register***)**
Specifies the address of the request parameter list that contains information to be used during the processing of the APPCCMD macroinstruction.

RPL and RPL Extension Fields Modified by Macroinstruction

Following are descriptions of RPL and RPL extension fields:

CONSTATE
The field in the RPL6 extension that indicates the state of the conversation. This field is labeled RPL6CCST in the RPL extension.

For half-duplex conversations, this field can have the following values:
X'01' SEND
X'02' RECEIVE
X'03' RECEIVE_CONFIRM
X'04' RECEIVE_CONFIRM_SEND
X'05' RECEIVE_CONFIRM_DEALLOCATE
X'06' PENDING_DEALLOCATE
X'07' PENDING_END_CONVERSATION_LOG
X'08' END_CONVERSATION
X'09' PENDING_SEND
X'0A' PENDING_RECEIVE_LOG

For full-duplex conversations, this field can have the following values:
X'80' FDX_RESET
X'81' SEND/RECEIVE
X'82' SEND_ONLY
X'83' RECEIVE_ONLY
X'84' PENDING_SEND/RECEIVE_LOG
X'85' PENDING_RECEIVE-ONLY_LOG
X'86' PENDING_RESET_LOG

FDB2
The field in the RPL in which a global VTAM secondary return code is returned to the application program. It is labeled RPLFDB2 in the RPL.

FMH5LEN
The field in the RPL extension that returns the length of the FMH-5 waiting to be received by the application program. If multiple FMH-5s are waiting to be received, FMH5LEN specifies the length of the longest FMH-5 to be received by the application program. This field has meaning only when FMH5RCV=YES. This field is labeled RPL6MH5L in the RPL extension.

FMH5RCV
The field in the RPL extension that returns an indication of whether an FMH-5 has been received. The indication is either YES or NO (RPL6RMH5 set on or off). This field is labeled RPL6RMH5 in the RPL extension.

YES (B'1')
One or more FMH-5s have been received from partner LUs. The FMH5RCV field continues to be set to YES so long as an FMH-5 is waiting to be received by the application program. The application program must issue APPCCMD CONTROL=RCVFMH5 in order to receive an FMH-5.

NO (B'0')
No FMH-5s are waiting to be received by the application program.

RCPRI
The field in the RPL extension in which an APPCCMD-specific primary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCPR in the RPL extension.

RCSEC
The field in the RPL extension in which an APPCCMD-specific secondary return code is returned to the application program. This field has meaning only when RTNCD=X'00' and FDB2=X'0B'. This field is labeled RPL6RCSC in the RPL extension. The combination of the RCPRI and RCSEC fields indicates the result of the macroinstruction processing.

RECLEN
The field in the RPL that returns to the application program the actual size of the structure containing the status information VTAM placed in the AREA if the RCPRI,RCSEC fields equal X'0000', X'0000'. If the RCPRI,RCSEC fields equal X'002C', X'0008' RECLEN indicates the length of the status data structure, but because the receive buffer is not sufficient to contain the entire structure, none of the status data structure is returned to the application program. This field is labeled RPLRLEN in the RPL.

RTNCD
The field in the RPL in which a global VTAM primary return code is returned to the application program. This field is labeled RPLRTNCD in the RPL.

USERFLD
Specifies 4 bytes of user data that the application program requests be associated with a conversation. Whenever an APPCCMD completes, VTAM places in the USERFLD field of the RPL extension the 4 bytes that were supplied on the APPCCMD CONTROL=ALLOC macroinstruction (if the conversation was initiated by the local application program) or the APPCCMD CONTROL=RCVFMH5 macroinstruction (if the conversation was initiated by the remote application program). This field is labeled RPL6USR in the RPL extension.

Return Codes

The following (RCPRI, RCSEC) combinations can be returned to the application program when it issues this APPCCMD macroinstruction. See “Chapter 2. Return Codes” on page 585 for a description of these return codes.

RCPRI	RCSEC	Meaning
X'0000'	X'0000'	OK
X'0000'	X'0009'	REQUEST_TERMINATED_BY_END_OF_CONVERSATION
X'002C'	X'0002'	PARAMETER_ERROR—INVALID_CONVERSATION_ID
X'002C'	X'0008'	PARAMETER_ERROR—SUPPLIED_LENGTH_INSUFFICIENT
X'002C'	X'000C'	PARAMETER_ERROR—ZERO_EXIT_FIELD
X'002C'	X'000D'	PARAMETER_ERROR—ZERO_ECB_FIELD
X'002C'	X'000E'	PARAMETER_ERROR—REQUEST_INVALID_FOR_ADDRESS_SPACE
X'002C'	X'000F'	PARAMETER_ERROR—CONTROL_BLOCK_INVALID
X'002C'	X'0010'	PARAMETER_ERROR—INVALID_DATA_ADDRESS_OR_LENGTH
X'002C'	X'0011'	PARAMETER_ERROR—PREVIOUS_MACROINSTRUCTION_OUTSTANDING
X'002C'	X'0032'	PARAMETER_ERROR—UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD
X'0070'	X'0000'	TEMPORARY_STORAGE_SHORTAGE

RCPRI	RCSEC	Meaning
X'0078'	X'0000'	VTAM_INACTIVE_FOR_YOUR_ACB
X'007C'	X'0000'	REQUEST_ABORTED
X'0090'	X'0000'	APPLICATION_NOT_APPC_CAPABLE
X'00A0'	X'0002'	REQUEST_NOT_ALLOWED—REQUEST_BLOCKED
X'00A8'	X'0000'	ENVIRONMENT_ERROR_OS_LEVEL_DOES_NOT_SUPPORT_ REQUESTED_FUNCTION
X'00A8'	X'0001'	ENVIRONMENT_ERROR—SUSPEND_FAILURE
X'00A8'	X'0002'	ENVIRONMENT_ERROR—RESUME_FAILURE

ISTGAPPC

Purpose

This macroinstruction declares and sets a list of global variables to indicate which LU 6.2 options are supported by the installed release of VTAM.

Usage

ISTGAPPC can be invoked directly, or by either IFGRPL or IFGACB as an inner macroinstruction call. The global variables defined for ISTGAPPC are shown in Table 2.

To use the ISTGAPPC macroinstruction, the programmer must be familiar with the GBLA and SETA assembler language instructions, which are described in the assembler language publication for your operating system.

The use of ISTGAPPC is similar to the use of the ISTGLBAL macroinstruction. For details, refer to the description of ISTGLBAL in *z/OS Communications Server: SNA Programming*.

The variables defined by ISTGAPPC are available to the application program at assembly time. If you want the application program to check these values at execution time, you can use the function-list vector described in the *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

Each global variable is an arithmetic symbol that can be set to 0, 1, or 2. Following are the meanings for the global variables and the corresponding levels of support.

Global Variable
Support Level

X'00' No (Option is not supported.)
X'01' Yes (Option is supported.)
X'02' Pass-through (VTAM offers support for this function, but the application program must implement the function.)

Context

Input states are not applicable to this macroinstruction.

Syntax

```
►►──┬──────────┬──ISTGAPPC──────────────────────────────────────────►◄
    └─name─────┘
```

Comments

ISTGAPPC sets the following global variables:

Table 2. LU 6.2 Global Macro Variables Set by ISTGAPPC

Variable	Function Indicated	Support Level
&ISTGA01	Conversations between transaction programs at the same LU	No
&ISTGA02	Delayed session allocation	No
&ISTGA03	Immediate session allocation	Yes
&ISTGA04	Sync point services	Pass-through

Table 2. LU 6.2 Global Macro Variables Set by ISTGAPPC (continued)

Variable	Function Indicated	Support Level
&ISTGA05	Program reconnect	No
&ISTGA06	Reserved	No
&ISTGA07	Session-level LU-LU verification	Yes
&ISTGA08	User identifier verification	Pass-through
&ISTGA09	Program-supplied user identifier and password	Pass-through
&ISTGA10	User identifier authorization	Pass-through
&ISTGA11	Profile verification and authorization	Pass-through
&ISTGA12	Reserved	No
&ISTGA13	Profile pass-through	Pass-through
&ISTGA14	Program-supplied profile	Pass-through
&ISTGA15	Send persistent verification	Pass-through
&ISTGA16	Receive persistent verification	Pass-through
&ISTGA17	PIP data	Pass-through
&ISTGA18	Logging of data in system log	Pass-through
&ISTGA19	Flush LU's SEND buffer	Yes
&ISTGA20	LUW identifier	Pass-through
&ISTGA21	Prepare to receive	Yes
&ISTGA22	Long locks	Yes
&ISTGA23	Post on receipt with wait	Pass-through
&ISTGA24	Post on receipt with test for posting	No
&ISTGA25	Receive immediate	Yes
&ISTGA26	Test for request-to-send received	Yes
&ISTGA27	Data mapping	Pass-through
&ISTGA28	FMH application program data	Pass-through
&ISTGA29	Get attributes	Pass-through
&ISTGA30	Get conversation type	Pass-through
&ISTGA31	Mapped conversation LU services component	Pass-through
&ISTGA32	CHANGE_SESSION_LIMIT verb	Yes
&ISTGA33	MIN_CONTENTION WINNERS_TARGET parameter	Yes
&ISTGA34	RESPONSIBLE(TARGET) parameter	Yes
&ISTGA35	DRAIN_TARGET(NO) parameter	Yes
&ISTGA36	FORCE parameter	No
&ISTGA37	ACTIVATE_SESSION verb	No
&ISTGA38	DEACTIVATE_SESSION verb	No
&ISTGA39	LU parameter verbs	Yes
&ISTGA40	LU-LU session limit	No
&ISTGA41	Locally-known LU names	Yes
&ISTGA42	Uninterpreted LU names	Yes
&ISTGA43	Single-session reinitiation	Yes
&ISTGA44	Alternate code processing	No
&ISTGA45	Maximum RU size bounds	Yes
&ISTGA46	Session-level mandatory cryptography	Yes
&ISTGA47	Contention-winner automatic-activation limit	Yes
&ISTGA48	Queued allocation of a contention-winner session	Yes
&ISTGA49	Enhanced security (SAME)	Pass-through
&ISTGA50	Session-level selective cryptography	Yes
&ISTGA51	Conversation group support	Yes
&ISTGA52	ALLOCATE WHEN_SESSION_FREE verb	Yes
&ISTGA53	LU 6.2 full-duplex protocols	Yes
&ISTGA54	VTAM-to-application vector list	Yes

Table 2. LU 6.2 Global Macro Variables Set by ISTGAPPC (continued)

Variable	Function Indicated	Support Level
&ISTGA55	Queued RCVFMH5	Yes
&ISTGA56	High performance data transfer	Yes
&ISTGA57	APPCCMD SENDRCV	Yes
&ISTGA58	Intra-LU conversations	Yes
&ISTGA59	Password substitution	Pass-through
&ISTGA60	Extended security sense codes	Pass-through
&ISTGA61	DCE security services	Pass-through

ISTRPL6

Purpose

This macroinstruction obtains storage for the RPL extension at assembly time and initializes any fields included as parameters on the macroinstruction. Any fields without a default value and not explicitly included on the macroinstruction are set to 0.

Context

Input states are not applicable to this macroinstruction.

Syntax

```
                                                  (1)                 (1)
   (2)                                  ┌─,CD=IMMED──────┐  ┌─,CONMODE=CS──────┐
►►─ ──────────────────────ISTRPL6 ──┤                    ├──┤                  ├──►
    └─name─┘                          └─,CD=─┬─IMMED─┬─┘  └─,CONMODE=─┬─CS────┬─┘
                                             └─DEFER─┘                  ├─BUFFCA─┤
                                                                        ├─LLCA───┤
                                                                        └─SAME───┘

                                                  (1)                 (1)
                                       ┌─,FILL=LL───────┐  ┌─,LIST=NONE───────┐
►──┬──────────────────────────────────┤                    ├──┤                  ├──►
   │                          (1)     └─,FILL=─┬─LL────┬─┘  └─,LIST=─┬─NONE──┬─┘
   └─,CONXMOD=─┬─CA────┬──────────────────────┘             ├─ALL───┤
               ├─CS────┤                                    └─NOSESS┘
               └─SAME──┘

                   (1)
   ┌─,LOCKS=SHORT───┐
►──┤                ├──┬──────────────────────────────────────────────────────────►
   └─,LOCKS=─┬─SHORT─┬─┘  │                                   (1)
             └─LONG──┘    └─,LOGMODE=8-byte_logon_mode_name──────┘

►──┬──────────────────────────────────┬──────────────────────────────────────────►
   │                          (1)     │                                   (1)
   └─,LUAFFIN=─┬─APPL────┬──────────┘  └─,LUNAME=8-byte_lu_name──────┘
               └─NOTAPPL─┘
```

```
►──┬──────────────
```

```
►──┬──────────────
```

placed in logical-record-continue-any, buffer-continue-any, or continue-specific mode. This field is labeled RPL6CMOD in the RPL extension.

CONMODE=BUFFCA
Specifies that the conversation is to be placed in buffer-continue-any mode. It indicates that this conversation is to apply when APPCCMD CONTROL=RECEIVE, QUALIFY=ANY is issued and that the application program is to receive data independently of the logical-record format of the data. BUFFCA corresponds to FILL=BUFFER on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

CONMODE=CS
Specifies that the conversation is to be placed in continue-specific mode. It indicates that data is to be received from this conversation by the application program only if the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC. When the application program issues APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC, it must indicate whether the data is to be received in terms of the logical-record format of the data, or independently of the logical-record format of the data.

CONMODE=LLCA
Specifies that the conversation is to be placed in logical-record-continue-any mode. It indicates that this conversation is to apply when APPCCMD CONTROL=RECEIVE, QUALIFY=ANY is issued and that the application program is to receive data in terms of the logical-record format of the data. LLCA corresponds to FILL=LL on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

CONMODE=SAME
Specifies that the continuation mode of the conversation should remain unchanged after the completion of the APPCCMD macroinstruction using this RPL.

CONXMOD
Specifies the mode for receiving expedited information upon completion of the APPCCMD.

CONXMOD=CS
Specifies that the mode for expedited information is to be put in such a state that expedited information can only be received by a specific-type of macroinstruction for such as, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC.

CONXMOD=CA
Specifies that the mode for expedited information is to be put in such a state that expedited information can only be received by either a specific-type of macroinstruction, for example, APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC or ISPEC, or by any type of macroinstruction, for example, APPCCMD CONTROL=RCVEXPD, QUALIFY=ANY or IANY.

CONXMOD=SAME
Specifies that the conversation mode for expedited data is to remain unchanged at the completion of this macroinstruction.

FILL
Specifies whether the application program is to receive data in terms of the logical-record format of the data. This parameter corresponds to FILL=LL|BUFFER described in the LU 6.2 architecture. The field is ignored unless QUALIFY=SPEC. This field is labeled RPL6FILL in the RPL extension.

FILL=BUFF
Specifies the application program is to receive data independently of its logical-record format, up to the length specified by the AREALEN field of the RPL. FILL=BUFF corresponds to FILL=BUFFER on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

FILL=LL
Specifies the application program is to receive one logical record, or a portion of the logical record up to the length specified by the AREALEN field of the RPL. If only a portion of the logical record is received, the DATA_INCOMPLETE bit in the what-received field is set on. The remainder of the logical record is buffered by VTAM, and will be used to satisfy the next RECEIVE request. FILL=LL corresponds to FILL=LL on the RECEIVE_AND_WAIT verb, as described in the LU 6.2 architecture.

LIST
Specifies the amount of detail to be provided about LUs, modes, and sessions. The requested information is provided in a RESTORE structure and describes the LUs, modes, and sessions that have been restored. This field is labeled RPL6LIST in the RPL extension.

LIST=ALL
Specifies that all LU, mode, and session information is included in the RESTORE structure.

LIST=NONE
Specifies that no RESTORE structure is returned.

LIST=NOSESS
Specifies that all LU and mode information is included in the RESTORE structure; session information is not included.

LOCKS
Specifies when the execution of the macroinstruction is complete following execution of the CONFIRM function. This field corresponds to the LOCKS parameter on the PREPARE_TO_RECEIVE verb as described in the LU 6.2 architecture. This field is labeled RPL6LOCK in the RPL extension.

LOCKS=SHORT
Specifies that the function of this macroinstruction is complete when a positive response is received to the confirmation request.

LOCKS=LONG
Specifies that the function of this macroinstruction is complete when information, such as data, is received from the partner LU following an affirmative reply to the confirmation request. The application program must issue an APPCCMD CONTROL=RECEIVE in order to get the information that caused the prior macroinstruction to complete.

LOGMODE=*8-byte_logon_mode_name*
The field that holds the logon mode name of the session over which an FMH-5 flows. It is an 8-byte name, padded on the right with blanks. This field is labeled RPL6MODE in the RPL extension.

LUAFFIN
Specifies whether the application program or VTAM will be the owner of the Generic Resource affinity for this specific LU partner.

LUAFFIN=APPL
The application program will own the GR affinity for this LU.

LUAFFIN=NOTAPPL
VTAM will own the GR affinity for this LU.

The LUAFFIN keyword is only meaningful when the issuing application is acting as a generic resource. If the application does not support a generic name, LUAFFIN is ignored.

The LUAFFIN value is honored if no sessions currently exist with the partner LU. If any active or pending sessions exist, the LUAFFIN value is ignored, and the previously established ownership is used for new sessions. If LUAFFIN is not specified and no sessions currently exist with the partner LU, the generic resource affinity ownership will be based on the type of LU 6.2 session or the owner will be the application if SETLOGON OPTCD=GNAMEADD, AFFIN=APPL was issued.

For more information about affinity ownership between an LU and a generic resource member, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

LUNAME=*8-byte_lu_name*
The field that holds the name of a partner LU. This LU name is the network name of the partner LU. It is an 8-byte name, padded on the right with blanks. This field is labeled RPL6LU in the RPL extension.

NETID=*8-byte_network_identifier*
The field that holds the network identifier of the partner LU. This identifier is the network identifier of the partner LU. If NQNAMES=YES, LUNAME and NETID are used together to form the network-qualified of the target LU. (If NETID is specified, LUNAME is specified.)

The network identifier is an 8-byte name, padded on the right with blanks. This field is labeled RPL6NET in the RPL extension.

QUALIFY=*one_of_the_qualify_values_listed_below*
Specifies the exact function of an APPCCMD macroinstruction. The general function of the macroinstruction is determined by the CONTROL keyword, required on each APPCCMD macroinstruction.

See the individual macroinstruction descriptions for details.

ABNDPROG
Specifies abnormal termination of a conversation because of a transaction program error.

ABNDSERV
Specifies abnormal termination of a conversation because of an LU services component error.

ABNDTIME
Specifies abnormal termination of a conversation because of excessive elapsed time.

ABNDUSER
Specifies abnormal termination of a conversation because of a user-specified condition.

ACTSESS
Responds positively to a session-initiation request being processed in the LOGON or SCIP exit.

ALL
Specifies a TESTSTAT that can return status on information that is available on any conversation.

ALLOCD
Allocates a session for use by a conversation.

ANY
Used to specify a RECEIVE or RCVEXPD that will accept normal or expedited information, respectively, for more than one conversation.

CNOS
Regulates session limits with another application program.

CONFIRM
Sends a confirmation request to another application program.

CONFRMD
Sends a reply to a confirmation request.

CONV
Deallocates the conversation and its underlying session.

CONVGRP
Associates a session having a specified conversation group identifier with a conversation for allocation of a conversation or deactivation of the session.

CONWIN
Allocates a conversation to a contention-winner session.

DACTSESS
Responds negatively to a session-initiation request in the LOGON or SCIP exit.

DATA
Sends data to a partner LU.

DATACON
Sends data and a confirmation request to a partner LU.

DATAFLU
Sends data to a partner LU and forces flushing of the SEND buffer.

DATAQUE
Specifies that the macroinstruction be queued pending receipt of the FMH-5 from the partner LU and that the FMH-5 as well as any data should be received to the application's buffer when received by VTAM.

DEFINE
Alters information in the LU-mode table.

DISPLAY
Displays information in the LU-mode table.

ERROR
Sends an error indication to a partner LU.

FLUSH
Forces flushing of the SEND buffer.

IALL
Specifies a TESTSTAT that can return status on information that is immediately available on any conversation.

IANY
Specifies a RECEIVE or RCVEXPD that can receive normal or expedited information, respectively, that is immediately available from a conversation in continue-any mode.

IMMED
Allocates a contention-winner session for immediate use by a conversation.

ISPEC
Specifies a RECEIVE that will accept normal information that is immediately available from a user-specified conversation.

NULL
Optional value that can be used when no other QUALIFY value applies

QUEUE
Specifies that the macroinstruction be queued pending receipt of the FMH-5 from the partner LU and that the FMH-5 should be received to the application's buffer when received by VTAM.

RESTORE
Restores a mode (or modes) that has been retained pending recovery of one or more persistent LU-LU sessions.

RESUME
Releases a session that has been suspended.

RQSEND
Requests that an application program be placed in SEND state.

SESSION
Deactivates the session and deallocates any conversation associated with it.

SPEC
Satisfies a RECEIVE using data for a particular conversation.

SUSPEND
Suspends a subsequent conversation.

SYNCBEG
Indicates the beginning of a synchronization exchange.

SYNCEND
Indicates the end of a synchronization exchange.

WHENFREE
Specifies to allocate a session for the conversation if a session is available or pending or one can be activated.

RTSRTRN
Specifies, upon completion of the APPCCMD, the manner in which Request_To_Send_Received indication is to be received.

RTSRTRN=BOTH
Specifies that Request_To_Send_Received indication can be received either by an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD or reported in the SIGRCV and SIGDATA fields returned with other APPCCMDs.

RTSRTRN=EXPD
Specifies that Request_To_Send_Received indication can be received only by an APPCCMD CONTROL=SENDEXPD or an APPCCMD CONTROL=RCVEXPD.

SENSE=*32-bit_unbind_sense_code*

SENSE=(*32-bit_unbind_sense_code_register***)**
The field that holds a 32-bit sense code. This field is labeled RPL6SNSO in the RPL extension.

TYPE
Specifies the level of error being reported on an APPCCMD CONTROL=SEND, QUALIFY=ERROR macroinstruction. This field is intended to distinguish between errors to be reported to end-user transaction programs and errors to be reported to a service component, such as a mapped conversation component, of the LU. This field is labeled RPL6TYPE in the RPL extension. See "APPCCMD CONTROL=SEND, QUALIFY=ERROR" on page 468 for more details.

TYPE=PROGRAM
Specifies an end-user transaction program error is being reported.

TYPE=SERVICE
Specifies a service-component error is being reported.

TYPE=USER
Specifies that the application program is providing to VTAM a user-specific sense code that it requests be placed in the FMH-7 that VTAM creates as a result of this APPCCMD macroinstruction.

USERFLD=*4_bytes_of_user_data*

USERFLD=(*user_data_register***)**
Specifies 4 bytes of user data that the application program requests be associated with a conversation. This field is labeled RPL6USR in the RPL extension.

VTRINA=*vector_address_field*

VTRINA=(*vector_address_register***)**
Specifies the address of the data area where VTAM places vector list information for the application.

This parameter is ignored if one of the following is true:

- VTRINA=0
- The value for VTRINL is less than the minimum length required to return the APPCCMD vector area header.
- The value for VTRINL is not specified.

This field is labeled RPL6VAIA in the RPL extension.

VTRINL=*vector_length_field*

VTRINL=(*vector_length_register***)**
Specifies the length of the data area where VTAM places vector list information for the application.

This parameter is ignored if the value for VTRINA is 0 or is not specified. This field is labeled RPL6VAIL in the RPL extension.

VTROUTA=*vector_address_field*

VTROUTA=(*vector_address_register***)**
Specifies the address of the area where the application places vector list information for VTAM. If OPTCD=XBUFLST is specified, this field must point to

the XBUFLST-receive vector (ISTAPC82), which is mapped by ISTAPCVL. (Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for more information.)

This field is labeled RPL6VAOA in the RPL extension.

VTROUTL=*vector_length_field*

VTROUTL=(*vector_length_register***)**
Specifies the length of the area where the application places vector list information for VTAM. This field is labeled RPL6VAOL in the RPL extension.

Chapter 2. Return Codes

VTAM passes feedback return codes to the LU 6.2 application program in a variety of ways. The principal feedback mechanism is the RCPRI and RCSEC return code fields in the RPL extension. These fields have meaning only when register 15 is set to X'00' and register 0 is set to X'0B'. These values are also the values of the RPL's RTNCD and FDB2 fields, respectively.

For a general discussion of how register contents relate to RPL feedback fields, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide*.

RCPRI and RCSEC Codes

The RPL extension contains two fields in which return code information is passed to the application program at the completion of an APPCCMD macroinstruction execution. The two fields are RPL6RCPRI and RPL6RCSEC, and together they indicate the result of the macroinstruction execution, including any state changes to the specified conversation. The RCPRI field returns a primary return code to the application; the RCSEC field returns a secondary return code to the application. Some RCPRI codes do not have associated RCSEC subcodes. For these RCPRI codes, the RCSEC field is set to X'0000'.

Some of the (RCPRI, RCSEC) return codes indicate the results of the local VTAM's processing of the macroinstruction; these return codes are returned on the APPCCMD that invoked the local processing. Other (RCPRI, RCSEC) return codes indicate the results of processing invoked at the remote end of the conversation and, depending upon the CONTROL and QUALIFY settings of the APPCCMD, can be returned on the APPCCMD that invoked the remote processing or on a subsequent APPCCMD. Still other return codes report events that originate at the remote end of the conversation.

The RCPRI and RCSEC codes are described below. Each description includes the meaning of the code, the reason for the condition indicated by the code, when the code can be reported to the application program, and the state of the conversation (if applicable) when the function of the APPCCMD completes. Actions taken by the local application program are discussed in the following return code descriptions in terms of APPCCMD macroinstructions; actions taken by the remote LU or transaction program are described more generically using the architected protocol boundary verbs documented in the LU 6.2 architecture.

Note: Some application programs change the hexadecimal values from the RCPRI, RCSEC fields to decimal values. You may need to convert these back to hexadecimal values for problem determination.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	(all)	USF6OK	OK

The local application program issued an APPCCMD macroinstruction that executed without error. The function defined for the APPCCMD was performed as specified.

The OK RCPRI code together with one of the following RCSEC subcodes form the complete return code that is returned to the application; the RCSEC subcode provides additional information.

© Copyright IBM Corp. 1988, 2001

RCPRI,RCSEC Combinations

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0000'	USF6OKSC	OK

The APPCCMD completed successfully and no additional information is defined for the APPCCMD. If a conversation-related macroinstruction is issued, the conversation state can be found in the CONSTATE field. Whenever this RCPRI,RCSEC combination is present, registers 15 and 0 are also set to 0.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0001'	USF6ASSP	AS SPECIFIED

The CNOS values supplied by the application program on the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction were accepted by the partner LU as specified, without negotiation.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0002'	USF6ASNG	AS NEGOTIATED

One or more of the CNOS values supplied by the application program on the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was changed by negotiation with the partner LU. The values are returned to the application program on the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction. (The macroinstruction description defines which values can be negotiated.)

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0003'	USF6RCVR	RECEIVE SPECIFIC REJECTED

An APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC or APPCCMD CONTROL=RECEIVE, QUALIFY=ISPEC macroinstruction was rejected because an APPCCMD CONTROL=RECEIVE, QUALIFY=ANY or APPCCMD CONTROL=RECEIVE, QUALIFY=IANY macroinstruction is currently being processed on this conversation. There is no state change. See Chapter 1. LU 6.2 Macroinstruction Syntax and Operands for more information on the APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC and APPCCMD CONTROL=RECEIVE, QUALIFY=ANY|IANY macroinstructions.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0004'	USF6SNGL	PARTNER LU SUPPORTS SINGLE SESSION

VTAM has determined that the partner LU supports only single sessions. If the session limit you specified was greater than 1, or if you did not specify a session limit, then the default values of 1, 0, 0 were used for your CNOS request.

If the partner LU indicated single-session capability using a negative BIND response, the partner LU's name will be missing from the Userdata subfield of the BIND. When the application program issues an APPCCMD CONTROL=OPRCNTL, QUALIFY=DISPLAY macroinstruction, it should verify the presence of the partner LU's fully qualified name. If the FQNLEN field is 0, the partner LU's name is not available. Check the FQNLEN field before checking the FQNAME field.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0005'	USF6INER	INTERNAL VTAM ERROR

VTAM rejected the APPCCMD CONTROL=REJECT, QUALIFY=SESSION macroinstruction because of an internal error other than a storage shortage condition.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0006'	USF6RSUN	RESTORE_UNNECESSARY—NO_MODES_TO_RESTORE

The APPCCMD CONTROL=OPRCNTL,QUALIFY=RESTORE macroinstruction is unnecessary. The associated mode (or modes) has been restored already, or nothing existed to restore.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'0007'	USF6RSIN	RESTORE_INCOMPLETE—INPUT_WORK_AREA_TOO_SMALL

The APPCCMD CONTROL=OPRCNTL,QUALIFY=RESTORE macroinstruction is incomplete. The AREA supplied is too small to hold all the information that needs to be returned. Reissue the macroinstruction one or more times to obtain all the restore information and to complete the restore.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'0000'	X'0008'	USF6NINA	NO IMMEDIATELY AVAILABLE INFORMATION

An APPCCMD that requested the immediate return of available information was issued. However, no information that could satisfy the request was available.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'0000'	X'0009'	USF6RTEC	REQUEST TERMINATED BY END OF CONVERSATION

An APPCCMD was awaiting processing or awaiting the arrival of information or a response on a specific conversation. The command has terminated because the conversation ended before the requested information became available or before it could be processed.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'0000'	X'000A'	USF6ANMS	SESSIONS WILL USE APPL NAME, GENERIC NAME REQUESTED

Use of the generic resource name was requested but the application network name is required.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'0000'	X'000B'	USF6GNMS	SESSIONS WILL USE GENERIC NAME, APPL NAME WAS REQUESTED

Use of the application network name was requested but the generic resource name is required.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'000C'	USF6NAM1	AS SPECIFIED, PARTNER LU KNOWN BY DIFFERENT NAME

The CNOS values supplied by the application program on the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction were acceptable by the partner LU as specified, without negotiation. Furthermore, the CNOS operation caused an LU entry of type RCVD_NAME to be changed to a VARIANT_NAME entry in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0000'	X'000D'	USF6NAM2	AS NEGOTIATED, PARTNER LU KNOWN BY DIFFERENT NAME

One or more of the CNOS values supplied by the application program on the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was changed by negotiation with the partner LU. The values are returned to the application program on the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction. (The macroinstruction description defines which values can be negotiated.) Furthermore, the CNOS operation caused an LU entry of type RCVD_NAME to be changed to a VARIANT_NAME entry in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	(all)	USF6ALLC	ALLOCATION ERROR

The application program issued APPCCMD CONTROL=ALLOC and allocation of the specified conversation could not be completed. When the ALLOCATION_ERROR RCPRI code is used with one of the following RCSEC subcodes (X'0000'–X'000F'), they form the complete return code that is returned to the program. The RCSEC subcode identifies the specific error. (The partner LU and remote transaction program referred to in the following RCSEC definitions are the

LU named in the LUNAME field of the APPCCMD, and the transaction program named in the FMH-5 supplied through the AREA field of the APPCCMD, respectively.)

If the partner LU detects the error that causes an ALLOCATION_ERROR RCPRI code to be returned to the application, the error indicator sent by the partner LU can specify that error log data follows the error indicator. The error log data indicator is returned to the application program in the LOGRCV field of the completed macroinstruction. If an ALLOCATION_ERROR RCPRI code is returned to the application along with LOGRCV=YES, the conversation should issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC to receive the error log data. When the error log data is received, the conversation is over.

If an ALLOCATION_ERROR RCPRI code is returned to the application along with LOGRCV=NO, the conversation is in END_CONV state.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0000'	USF6ALNR	ALLOCATION FAILURE, NO RETRY

The conversation cannot be allocated on a session because of a permanent condition. For example, the session to be used for the conversation cannot be activated for one of the following reasons:

- The mode is closed; the current session limit is 0.
  - CNOS has not been negotiated and no entry has been created for the mode.
  - A previous CNOS request has set limits to 0.
- A system definition error.
- A session-activation protocol error.

The session also might be deactivated because of a session protocol error before the conversation could be allocated. The application program should not retry the allocation request until the condition is corrected. The application should check the returned SENSE field in the RPL extension for an indication of the exact error.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0001'	USF6ALR	ALLOCATION FAILURE RETRY

The conversation cannot be allocated on a session because of a temporary condition. For example, the session to be used for the conversation cannot be activated because of a temporary lack of resources at the remote LU; or the session was deactivated because of session outage before the conversation could be allocated. The condition is temporary, and the program can retry the allocation request.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0002'	USF6ALCM	CONVERSATION TYPE MISMATCH

The partner LU rejected the allocation request because the remote transaction program does not support the respective mapped or basic protocol boundary. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0003'	USF6ALPI	PIP NOT ALLOWED

The partner LU rejected the allocation request because the local application program provided program initialization parameter (PIP) data (along with the FMH-5) and either the partner LU does not support PIP data, or the remote transaction program has no PIP variables defined. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0004'	USF6ALPP	PIP NOT SPECIFIED CORRECTLY

The partner LU rejected the allocation request because the remote transaction program has one or more PIP variables defined and the local application program provided no program initialization parameters, or the local application program specified program initialization parameters (along with the FMH-5) that do not correspond in number to those defined for the remote transaction program. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0005'	USF6ALSC	SECURITY NOT VALID

The partner LU rejected the allocation request because the access security information supplied by the local application (in the FMH-5) is not valid. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0006'	USF6ALSY	SYNC LEVEL NOT SUPPORTED BY LU

The partner LU rejected the allocation request because the synchronization level specified in the allocation request is not supported by both the local and partner LU. The local LU specifies its level of synchronization support on its APPL statement. The partner LU has returned the negotiated level between the two LUs in the BIND response. This return code is returned on the APPCCMD CONTROL=ALLOC macroinstruction for the local LU.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0007'	USF6ALSL	SYNC LEVEL NOT SUPPORTED BY PROGRAM

The partner LU rejected the allocation request because the local application program specified a synchronization level (in the FMH-5) that the remote transaction program does not support. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0008'	USF6ALTP	TPN NOT RECOGNIZED

The partner LU rejected the allocation request because the local application program specified a remote transaction program name (TPN) that the partner LU does not recognize. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'0009'	USF6ALTN	TRANSACTION PROGRAM NOT AVAILABLE, NO RETRY

The partner LU rejected the allocation request because the local application program specified a remote transaction program that the partner LU recognizes but cannot start. The condition is not temporary, and the application should not retry the allocation request. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'000A'	USF6ALTR	TRANSACTION PROGRAM NOT AVAILABLE, RETRY

The partner LU rejected the allocation request because the local application specified a remote program that the remote LU recognizes but currently cannot start. The condition is temporary, and the application can retry the allocation request. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'000B'	USF6ALRN	CANNOT RECONNECT TRANSACTION PROGRAM, NO RETRY

The partner LU rejected the reconnection request because it does not recognize the conversation correlator. The condition is not temporary, and the application should not retry the reconnection request. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'000C'	USF6ALRR	CANNOT RECONNECT TRANSACTION PROGRAM, RETRY

The partner LU rejected the reconnection request because it currently cannot reconnect the remote transaction program implied by the conversation correlator. The condition is temporary, however, and the application can retry the reconnection request. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'000D'	USF6ALNS	RECONNECT NOT SUPPORTED BY PROGRAM

The partner LU rejected the allocation request because the local application program specified a recovery level of program reconnect (in the FMH-5) and the remote transaction program does not support program reconnect. This return code is returned on an APPCCMD subsequent to APPCCMD CONTROL=ALLOC.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'000E'	USF6SPMA	MODE MUST BE RESTORED BEFORE USING

The APPCCMD CONTROL=ALLOC macroinstruction is rejected because the specified mode name is pending recovery for persistent LU-LU sessions. Restore the mode by issuing APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0004'	X'000F'	USF6DARQ	DEALLOCATION REQUESTED

The allocation request has been canceled before its normal processing could be completed. The local application program issued a request for abnormal deallocation of the pending conversation.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'0004'	X'0010'	USF6ALSF	ALLOCATION ERROR - SYNCH LEVEL NOT VALID FOR FULL-DUPLEX

The allocation request has been rejected because it specifies a full-duplex conversation with a sync point level not allowed for a full-duplex conversation.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'0004'	X'0011'	USF6LNSF	ALLOCATION ERROR - LU PAIR NOT SUPPORTING FDX CONVERSATION

The allocation request has been rejected because it specifies a full-duplex conversation and the negotiated level of support between the local application and the partner LU does not allow full-duplex conversations.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	(all)	USF6CNSA	CNOS FAILURE

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction did not process successfully. The CNOS_ALLOCATION_ERROR RCPRI code together with

one of the following RCSEC subcodes (X'0000'–X'0006') form the complete return code that is returned to the transaction program. The RCSEC subcode identifies the specific error. The local and partner LUs' CNOS parameters are not changed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0000'	USF6CANR	ALLOCATION FAILURE, NO RETRY

The control operator conversation cannot be allocated because of a condition that is not temporary. For example, the session to be used for the control operator conversation cannot be activated because the session limit for the specified partner LU and SNASVCMG mode name is currently 0 at either the local LU or partner LU; or because of a system definition error or a session-activation protocol error; or because a session protocol error caused the session to be deactivated before the conversation could be allocated. The CNOS will not be able to complete successfully until the condition is corrected. This code can also be returned if a partner LU rejects a SNASVCMG mode name BIND.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0001'	USF6CAR	ALLOCATION FAILURE, RETRY

The control operator conversation cannot be allocated because of a temporary condition. For example, the session to be used for the control operator conversation cannot be activated because of a temporary lack of resources at the local LU or partner LU, or the session was deactivated because of session outage before the conversation could be allocated. The condition is temporary, and the control operator can retry the transaction later.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0002'	USF6CATR	TRANSACTION PROGRAM NOT AVAILABLE, RETRY

The partner LU is currently unable to start the transaction program identified as hex 06F1, which is the SNA service transaction program for the control operator. For example, there can be a temporary lack of resources the partner LU needs to start the transaction program. The condition is temporary, and the control operator can retry the transaction later.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0003'	USF6CATN	TRANSACTION PROGRAM NOT AVAILABLE, NO RETRY

The partner LU is unable to start the transaction program identified as X'06F1', which is the SNA service transaction program for the control operator. The condition is not temporary, and the application should not retry the CNOS request.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0004'	USF6CACM	CONVERSATION TYPE MISMATCH

The partner LU rejected the CNOS conversation allocation request because the remote transaction program does not support the respective mapped or basic protocol boundary.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0005'	USF6CASC	SECURITY NOT VALID

The partner LU rejected the CNOS conversation allocation request because the access security information supplied by VTAM (in the FMH-5) is not valid.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0006'	USF6SPMC	MODE MUST BE RESTORED BEFORE USING

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction is rejected because the specified mode name is pending recovery for persistent LU-LU sessions. Restore the mode by issuing APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE. New modes can be added once the SNASVCMG mode for an LU has been restored, but any mode that exists when the failure (or takeover) occurs cannot be used until that mode has been restored.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0008'	X'0007'	USF6NQNM	NETWORK QUALIFIED NAME MISMATCH

The name on an APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was an ACB name. The ACB name is not identical to the network resource name. ACB names cannot be used in cross-domain, cross-network, or network qualified. For information on coding the ACBNAME operand, see the *z/OS Communications Server: SNA Resource Definition Reference*.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'000C'	X'0000'	USF6CNSN	CNOS RESOURCE FAILURE, NO RETRY

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction did not execute successfully because of a failure that caused the control operator conversation to be deallocated prematurely. For example, the session being used for the control operator conversation was deactivated for one of the following reasons:

- A session protocol error
- A session outage from which the control operator component of the LU could not recover

The conversation also might be deallocated because of a protocol error between the control operator components of the LUs. The condition is not temporary, and the control operator should not retry the transaction until the condition is corrected. The CNOS parameters remain unchanged at the local LU, or both the local and partner LUs, depending on when the failure occurred.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0010'	(all)	USF6CRRJ	COMMAND RACE REJECT

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction did not execute successfully because two CNOS operations caused contention for the needed resources.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0010'	X'0000'	USF6CRPR	PARTNER GRANTED RETRY

Both LUs initiated a CNOS negotiation for the same mode at the same time. The partner LU will retry the CNOS request. VTAM fails the CNOS request from the local LU.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0010'	X'0001'	USF6CRLR	CONTROL OPERATOR FOR LOCAL LU RETRIED

Both LUs initiated CNOS processing for the same mode at the same time. VTAM failed the partner's CNOS attempt, and the local LU was given permission to retry the CNOS request. VTAM attempted CNOS processing again but the subsequent CNOS negotiation failed as well. VTAM was forced to fail the local LU's CNOS request.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0010'	X'0002'	USF6PCIP	PARTNER CNOS IN PROGRESS

The partner LU has already begun processing a CNOS for the same mode name, and its processing will continue uninterrupted. The application program must reissue this APPCCMD for it to be processed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0010'	X'0003'	USF6LPSS	LU IN PENDING SINGLE STATE

The CNOS negotiation cannot be attempted at this time because the partner LU has initiated a CNOS request for the same mode. The partner LU might be a single-session-capable LU. The local LU cannot issue a CNOS request until the CNOS request initiated by the partner LU completes.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0010'	X'0004'	USF6PLSS	PARTNER LU STARTING SESSION

A partner LU that provides only single-session support is currently initiating a session. Because only one session can be active at a time, the application

program's CNOS request is rejected. The application program can retry the CNOS command later.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0014'	X'0000'	USF6DABP	DEALLOCATE ABEND PROGRAM

The remote transaction program issued a DEALLOCATE verb, as defined in the LU 6.2 architecture, specifying the TYPE(ABEND_PROG) parameter, or the remote LU did so because of a remote transaction program abend condition. If the conversation for the remote transaction program was in a state in which information can be received when the DEALLOCATE was issued, information sent by the local application and not yet received by the remote transaction program was purged. This return code can be reported to the local application on any APPCCMD macroinstruction that can process the error notification on a half-duplex conversation. This return code can only be reported on an APPCCMD CONTROL=RECEIVE on a full-duplex conversation. The error indicator sent by the partner LU to specify the DEALLOCATE_ABEND_PROGRAM condition can specify that error log data follows the error indicator. The error log data indicator is returned to the application program in the LOGRCV field of the completed macroinstruction. If a DEALLOCATE_ABEND_PROGRAM RCPRI code is returned to the application along with LOGRCV=YES, the conversation should issue APPCCMD CONTROL=RECEIVE, QUALIFY=SPEC|ISPEC to receive the error log data. The conversation is then ended. If a DEALLOCATE_ABEND_PROGRAM RCPRI code is returned to the application along with LOGRCV=NO, the conversation is ended.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0018'	X'0000'	USF6DABS	DEALLOCATE ABEND SERVICE

The remote transaction program issued a DEALLOCATE verb, as described in the LU 6.2 architecture, specifying the TYPE(ABEND_SVC) parameter. If the conversation for the remote transaction program was in a state in which information can be received when the DEALLOCATE was issued, information sent by the local application and not yet received by the remote transaction program was purged. This return code can be reported to the local application on any APPCCMD macroinstruction that can process the error notification on a half-duplex conversation. This return code can only be reported on an APPCCMD CONTROL=RECEIVE on a full-duplex conversation. The error indicator sent by the partner LU to specify the DEALLOCATE_ABEND_SERVICE condition can specify that error log data follows the error indicator. The error log data indicator is returned to the application program in the LOGRCV field of the completed macroinstruction. If a DEALLOCATE_ABEND_SERVICE RCPRI code is returned to the application along with LOGRCV=YES, the conversation is in PEND_END_CONV_LOG or PEND_RESET_LOG state. If a DEALLOCATE_ABEND_SERVICE RCPRI code is returned to the application along with LOGRCV=NO, the conversation is in END_CONV or FDX_RESET state.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'001C'	X'0000'	USF6DABT	DEALLOCATE ABEND TIMER

The remote transaction program issued a DEALLOCATE verb, as described in the LU 6.2 architecture, specifying the TYPE(ABEND_TIMER) parameter. If the conversation for the remote program was in a state in which information can be received when the DEALLOCATE was issued, information sent by the local application program and not yet received by the remote transaction program was purged. This return code can be reported to the local program on any APPCCMD macroinstruction that can process the error notification on a half-duplex conversation. This return code can only be reported on an APPCCMD CONTROL=RECEIVE on a full-duplex conversation. The error indicator sent by the partner LU to specify the DEALLOCATE_ABEND_TIMER condition can specify that error log data follows the error indicator. The error log data indicator is returned to the application program in the LOGRCV field of the completed macroinstruction. If a DEALLOCATE_ABEND_TIMER RCPRI code is returned to the application along with LOGRCV=YES, the conversation is in PEND_END_CONV_LOG or PEND_RESET_LOG state. If a DEALLOCATE_ABEND_TIMER RCPRI code is returned to the application along with LOGRCV=NO, the conversation is in END_CONV or FDX_RESET state.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0020'	X'0000'	USF6CNSR	CNOS FAILURE, RETRY

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued and a conversation was begun with the partner LU. However, 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 or a modem failure. The condition is temporary, and the application can retry the transaction.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0024'	X'0000'	USF6LRBE	LOGICAL RECORD BOUNDARY ERROR

The application program began sending a logical record before the previous logical record was sent in its entirety. The conversation state does not change.

For macroinstructions that use the QUALIFY=DATACON keyword, the data that was to be sent with the confirmation request is held. The application program must either furnish more data to finish the logical record, or truncate the incomplete record. The application cannot immediately send more data to complete the logical record, but must explicitly flush the send buffer and then send data to complete the logical record.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0028'	X'0000'	USF6SLCL	LU MODE SESSION LIMIT CLOSED

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction did not execute successfully because the partner LU currently will not allow the session limit for the specified mode name to be raised above 0. The session limit remains at 0. This condition is not necessarily permanent; the control operator can retry the CNOS transaction later.

RCPRI,RCSEC Combinations

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	(all)	USF6PARM	PARAMETER ERROR

VTAM rejected the APPCCMD because one of the RPL, RPL extension, or session limits structure fields specified in the APPCCMD contained a value that was not valid. The PARAMETER_ERROR RCPRI code together with the following RCSEC subcodes (X'0000'–X'002D') form the complete return code that is returned to the application. The subcode identifies the specific error. This RCPRI code is returned on the APPCCMD that contained the parameter that was not valid. When this RCPRI code is returned on a conversation APPCCMD macroinstruction (that is, a macroinstruction that does not specify CONTROL=OPRCNTL), the state of the conversation remains unchanged. When this RCPRI code is returned on an APPCCMD CONTROL=OPRCNTL macroinstruction, the local and partner LUs' CNOS parameters are not changed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0000'	USF6IVLU	INVALID LU NAME OR NETWORK IDENTIFIER

The APPCCMD specified an unrecognized partner LU name or network identifier.

This combination of return codes might result if VTAM does not find the LU name for a partner in the LU-mode table. The partner LU name and the logon mode name are added to the dynamically built LU-mode table during CNOS negotiation. To initiate CNOS negotiation, the application program issues the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction and specifies the LU name and logon mode (LOGMODE) name to be used during communication.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0001'	USF6IVMD	INVALID MODE

The APPCCMD specified an unrecognized logmode name, or the logmode name is not allowed for the LU-LU pair.

This combination of return codes might occur if the LU name specified for a conversation allocation request is present in the LU-mode table but the logon mode name is not present. The partner LU name and the logon mode name are added to the dynamically built LU-mode table during CNOS negotiation. To initiate CNOS negotiation, the application program issues the APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction and specifies the LU name and logon mode (LOGMODE) name to be used during communication.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0002'	USF6IVCI	INVALID CONVERSATION

The APPCCMD specified an unassigned conversation ID, or the RPL used for the request specified an ACB other than the one associated with the conversation assigned that CONVID. The value specified might have been a valid CONVID, but

the conversation might not be active.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0003'	USF6IVLL	INVALID LL

The data provided by the application program on an APPCCMD CONTROL=SEND, an APPCCMD CONTROL=PREPRCV, or an APPCCMD CONTROL=DEALLOC macroinstruction was not valid. It contained a logical record length (LL) value of X'0000', X'0001', X'8000', or X'8001'. An LL value of hex 0001, which indicates that the data contains a presentation services (PS) header for sync point, is allowed only on conversations with a synchronization level of sync point.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0004'	USF6IVSV	INVALID VALUES FOR SNASVCMG MODE

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued and the values specified for the SESSLIM, MINWINL, and MINWINR do not specify (2,1,1) or (0,0,0), respectively.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0005'	USF6IVDL	INVALID DRAINL CHANGE

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued, NBRMODE=ONE and DRAINL=YES were specified, the session limit in effect when the APPCCMD was issued was 0, and DRAINL=NO was in effect when the APPCCMD was issued. (The application program attempted to change DRAINL from NO to YES on an APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction when session limits were 0.)

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0006'	USF6SNAR	SNASVCMG MODE CANNOT CURRENTLY BE RESET

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction is issued, the SNASVCMG mode name is specified, and either one or more session limits for the mode name group for the partner LU is not 0; or one or more session limits for the mode name group for the partner LU is 0, but draining is enabled.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0007'	USF6MMEX	MINWINL PLUS MINWINR EXCEEDS SESSLIM

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS or QUALIFY=DEFINE macroinstruction was issued and either the sum of MINWINL plus MINWINR is greater than the SESSLIM value specified, or the sum of DMINWNL plus DMINWNR is greater than the DSESLIM value specified.

RCPRI,RCSEC Combinations

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0008'	USF6LNIN	SUPPLIED LENGTH INSUFFICIENT

The application issued one of the following macroinstructions:
- APPCCMD CONTROL=RCVEXPD
- APPCCMD CONTROL=RCVFMH5
- APPCCMD CONTROL=RECEIVE,OPTCD=XBUFLST
- APPCCMD CONTROL=OPRCNTL,QUALIFY=ACTSESS
- APPCCMD CONTROL=OPRCNTL,QUALIFY=DISPLAY
- APPCCMD CONTROL=OPRCNTL,QUALIFY=RESTORE
- APPCCMD CONTROL=TESTSTAT.

The data area or data length was not suitable as indicated in the following:

RECEIVE,OPTCD=XBUFLST
The area specified is not large enough to hold one extended buffer list entry.

RCVEXPD
Data area is too small to contain all the expedited data.

RCVFMH5
Data area is too small to contain the next available FMH-5.

QUALIFY=ACTSESS
Data length indicated in the supplied session parameters was larger than the amount of data provided or exceeds the maximum size allowed.

QUALIFY=DISPLAY
Data area is too small to contain the DEFINE/DISPLAY (ISTSLD) structure.

QUALIFY=RESTORE
Data area is too small to contain the RESTORE (ISTREST) structure.

TESTSTAT
Data area is too small to contain the status data structure (ISTSTATD).

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0009'	USF6INSL	INCOMPLETE STRUCTURE SUPPLIED

The application program issued one of the following macroinstructions:
- APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS
- APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS
- APPCCMD CONTROL=OPRCNTL, QUALIFY=DEFINE.

The data length was not suitable as indicated in the following:

QUALIFY=ACTSESS
Data length provided was less than the minimum size for the session parameters.

QUALIFY=CNOS
Data length provided was less than the minimum size for the session limits structure (ISTSLCNS).

QUALIFY=DEFINE

Data length provided was less than the minimum size for the DEFINE/DISPLAY (ISTSLD) structure.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'000A'	USF6INFM	INCOMPLETE FMH5 SUPPLIED

The application program issued APPCCMD CONTROL=ALLOC, but did not supply an entire FMH-5.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'000B'	USF6INGD	INCOMPLETE GDS VARIABLE SUPPLIED

The application program issued an abnormal termination APPCCMD deallocation macroinstruction, but did not supply an entire GDS variable.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'000C'	USF60EXT	ZERO EXIT FIELD

The RPL specified that the ECB-EXIT field is being used as an EXIT field, but the RPL exit routine address in the field is 0. No RPL exit routine has been scheduled.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'000D'	USF60ECB	ZERO ECB FIELD

The RPL specified that the ECB-EXIT field is being used to point to an external ECB, but the address in the field is 0. No ECB has been posted.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'000E'	USF6RIAS	REQUEST INVALID FOR ADDRESS SPACE

An internal error occurred.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'000F'	USF6CBIN	CONTROL BLOCK INVALID

The RPL's ACB field does not contain the address of a valid ACB or the ACB is closed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0010'	USF6INDL	INVALID DATA ADDRESS OR LENGTH

RCPRI,RCSEC Combinations

An APPCCMD was issued that specified a work area address that is beyond the addressable range of the application program.

If using a buffer list or extended buffer list to send data, check entries to ensure that the length field does not contain any negative values.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0011'	USF6PRVO	PREVIOUS MACROINSTRUCTION OUTSTANDING

An APPCCMD is issued that specifies a conversation resource while an outstanding macroinstruction that targets the same conversation and processes on the same conversation queue is pending completion, or an APPCCMD CONTROL=OPRCNTL is issued while an outstanding operator control APPCCMD that targets the same LU is pending completion. Wait until the first macroinstruction completes or coordinate this request with the one that is outstanding.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0012'	USF6BLIV	BUFFER LIST LENGTH INVALID

The RECLEN field of the RPL was not valid.

For the following macroinstructions, the RECLEN field must be a nonzero multiple of 16:
- APPCCMD CONTROL=DEALLOC, OPTCD=BUFFLST
- APPCCMD CONTROL=PREPRCV, OPTCD=BUFFLST
- APPCCMD CONTROL=SEND, OPTCD=BUFFLST
- APPCCMD CONTROL=SENDEXPD, OPTCD=BUFFLST
- APPCCMD CONTROL=SENDRCV, OPTCD=BUFFLST.

For the following macroinstructions, the RECLEN field must be a nonzero multiple of 48:
- APPCCMD CONTROL=DEALLOC, OPTCD=XBUFLST
- APPCCMD CONTROL=PREPRCV, OPTCD=XBUFLST
- APPCCMD CONTROL=SEND, OPTCD=XBUFLST

For the APPCCMD CONTROL=SENDRCV, OPTCD=XBUFLST macroinstruction, the value for RECLEN minus 16 must be a nonzero multiple of 48.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0013'	USF6NOMD	NO CORRESPONDING MODE IN LM TABLE

The application program issued one of the following macroinstructions:
- APPCCMD CONTROL=OPRCNTL, QUALIFY=DISPLAY
- APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE.

The application program also specified a mode name for which no corresponding entry exists in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0014'	USF6IVBP	INVALID BIND PARAMETERS

The application program issued an APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS and specified a set of BIND parameters that were not valid, or the parameters in the BIND that was received were not valid.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0015'	USF6IVTP	INVALID TPN

The application program issued an APPCCMD CONTROL=ALLOC with an FMH-5 that contained a transaction program name that was reserved or not valid, such as X'06F1', which is the SNA service transaction program for the control operator.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0016'	USF6NOLU	NO CORRESPONDING LU IN LM TABLE

The application program issued one of the following macroinstructions:
- APPCCMD CONTROL=OPRCNTL, QUALIFY=DISPLAY
- APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE.

The application program also specified an LU name for which no corresponding entry exists in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0017'	USF6IMDF	INVALID MODE SPECIFIED

The application program issued an APPCCMD CONTROL=OPRCNTL, QUALIFY=DEFINE macroinstruction and specified mode name SNASVCMG.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0018'	USF6ILSP	INVALID LIMIT SPECIFIED

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued and one of the session limit fields was an incorrect value.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0019'	USF6SMAI	SNASVCMG MODE ALREADY INITIALIZED

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued in order to initialize the SNASVCMG mode. However, it was already initialized, and no action was taken.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'001A'	USF6ALLS	ALL MODES SPECIFIED ON SINGLE SESSION LU

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued against all the mode names of the LU specified. However, the partner LU is single-session capable. Therefore, an APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction must be issued against a specific mode name.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'001B'	USF6SMSS	SNASVCMG OR CPSVCMG MODE FOR SINGLE SESSION LU

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued for the SNASVCMG or CPSVCMG mode name. However, the partner LU is single-session capable, and the SNASVCMG or CPSVCMG is not allowed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'001C'	USF6SSMI	SINGLE SESSION, MODE ALREADY INITIALIZED

An APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction was issued for a partner LU that is single-session capable. However, another of the LU's mode names is already initialized to nonzero session limits, and only one mode name can have nonzero session limits at a time.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'001E'	USF6CIDI	CID INVALID

The RPL's ARG field does not contain a valid session identifier (CID). You might have inadvertently modified the field or failed to set it in the first place, or you might have used the CID of a session that no longer exists.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'001F'	USF6APNA	APPCCMD ISSUED FOR NON-APPC

The application issued an APPCCMD against a non-LU 6.2 session or resource. The APPCCMD is rejected.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0020'	USF6PRRO	PREVIOUS REJECT REQUEST OUTSTANDING

An APPCCMD CONTROL=REJECT request was issued. However, a previous APPCCMD CONTROL=REJECT request has already been issued for the same resource. The later APPCCMD CONTROL=REJECT was rejected.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0021'	USF6DARJ	ABNORMAL DEALLOCATE REJECTED, RETRY

One of the following macroinstructions was issued:
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDPROG
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDSERV
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDTIME
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDUSER.

However, a prior macroinstruction that cannot be canceled is outstanding. The command is not allowed in this case and is rejected. This command also is not allowed to be issued when the conversation is in RECEIVE state and no data has been received for the conversation. APPCCMD CONTROL=REJECT, QUALIFY=CONV can be issued to terminate the conversation and session in this case.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0022'	USF6IVCQ	INVALID CONTROL OR QUALIFY VALUE

An undefined value for the CONTROL or QUALIFY keyword was specified, or a QUALIFY value is not valid to use with the specified CONTROL value. For CONTROL types that do not use a QUALIFY value, RPL6QUAL must be set to 0.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0023'	USF6INSI	INVALID SESSION INSTANCE IDENTIFIER

VTAM rejected an APPCCMD CONTROL=REJECT, QUALIFY=SESSION request or an APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND request or an APPCCMD CONTROL=SETSESS, QUALIFY=RESUME request because the local application specified:
- A session instance identifier for a session that was not active at the time of the request.
- A session ID length that was not valid.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0024'	USF6PSHI	PS HEADER NOT SUPPLIED

VTAM rejected the APPCCMD CONTROL=SEND request because the local application did not supply a complete PS header. (For example, the PS header length and data following are missing.)

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0025'	USF6PSLI	PS HEADER LENGTH IS INSUFFICIENT

VTAM rejected the APPCCMD CONTROL=SEND request because the local application specified an insufficient PS header length (the length equals 0).

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0026'	USF6NMSC	SESSION INSTANCE IDENTIFIER AND CONVERSATION IDENTIFIER MISMATCH

VTAM rejected the APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND request because the application program requested a session with APPCCMD CONTROL=SETSESS, QUALIFY=SUSPEND, but the conversation identified by CONVID was not currently assigned to the session identified by SESSID. VTAM rejected the request and nothing was suspended.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0027'	USF6IDET	INVALID DEACTIVATION TYPE CODE

VTAM rejected the APPCCMD CONTROL=REJECT, QUALIFY=SESSION request because the local application program omitted the DEACTYP parameter or specified an UNBIND deactivation type code value other than cleanup (X'0F') or protocol violation (X'FE'). The session has been successfully deactivated with UNBIND (X'0F').

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0028'	USF6NCRY	CRYPTOGRAPHY NOT ALLOWED ON MODE

An APPCCMD CONTROL=SEND, an APPCCMD CONTROL=PREPRCV, or an APPCCMD CONTROL=DEALLOC macroinstruction is rejected because CRYPT=YES is specified, and the mode does not support encryption.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0029'	USF6INLI	INVALID LIST VALUE SPECIFIED ON APPCCMD FOR RESTORE

The value for the LIST field in the RPL is not equal to NONE, ALL, or NOSESS. The keyword LIST=ALL, LIST=NONE, or LIST=NOSESS can be specified on the APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE macroinstruction.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'002A'	USF6INCG	INVALID CGID VALUE SPECIFIED

A macroinstruction was issued specifying CONVGRP, but the conversation group ID (CGID) was not valid. You might have unintentionally modified the field, failed to set it correctly, or used a CGID that corresponds to a session that no longer exists.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'002B'	USF6NONI	NETWORK-QUALIFIED NAME REQUIRED

NETID was not coded on the APPCCMD although PARMS=(NQNAMES=YES) was coded on the ACB macroinstruction.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'002C'	X'002C'	USF6INEL	PARAMETER ERROR - INVALID EXPEDITED DATA LENGTH

An APPCCMD CONTROL=SENDEXPD was issued that specified an expedited data length of 0 or an expedited data length greater than the allowed maximum. The largest expedited data size that can be sent with one macroinstruction invocation is 86 bytes.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'002C'	X'002D'	USF6INSC	PARAMETER ERROR - INVALID SENSE CODE VALUE SPECIFIED

An APPCCMD CONTROL=DEALLOC|DEALLOCQ,QUALIFY=ABNDUSER was specified with a sense code that was not an allocation or abnormal deallocation sense code value.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'002E'	USF6VANV	VECTOR AREA NOT VALID

The application supplied VTAM with a vector area address that is not valid or is write-protected.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'002F'	USF6VALI	VECTOR AREA LENGTH INSUFFICIENT

The application supplied VTAM with a vector area that is smaller than the minimum required size.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0030'	USF6STNV	PARAMETER_ERROR—STORAGE_TYPE_NOT_VALID

A storage type indication was not supplied or is not valid. Storage type is required to be specified via the ISTAPC82 mapping DSECT that is mapped within the

ISTAPCVL mapping DSECT.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0031'	USF6VALS	PARAMETER_ERROR—SENDRCV_SPECIFIED_WITHOUT_OPTCD=BUFFLST|XBUFLST

The APPCCMD CONTROL=SENDRCV was issued without specifying a buffer. OPTCD=BUFFLST|XBUFLST is required for this macroinstruction.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0032'	USF6UNXV	PARAMETER_ERROR—UNEXPECTED_VECTOR_PROVIDED_ON_APPCCMD

An unexpected vector was provided on an APPCCMD request. An input vector is not defined for the APPCCMD.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0033'	USF6VNPV	PARAMETER_ERROR—A_REQUIRED_VECTOR_WAS_NOT_PROVIDED_OR_SPECIFIED_INCORRECTLY

A required input vector was either not provided or specified incorrectly on an APPCCMD request.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'002C'	X'0034'	USF6LNSP	PASSWORD_SUBSTITUTION_VALUE_SET_IN_ERROR

The FMH-5 received from the application indicated password substitution in byte 4, bit 3. The session established with the partner does not support password substitution. Reissue the macroinstruction with this bit setting off.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0030'	X'0000'	USF6PENT	PROGRAM ERROR NO TRUNCATION

The remote transaction program issued an LU 6.2 SEND_ERROR verb specifying the TYPE(PROG) parameter; the conversation for the remote program was in a sending state; and the LU 6.2 SEND_ERROR verb did not truncate a logical record. No truncation occurs when a transaction program issues the LU 6.2 SEND_ERROR verb before sending any logical records or after sending a complete logical record. This return code is reported to the local application program when it issues an APPCCMD CONTROL=RECEIVE macroinstruction prior to receiving any logical records or after receiving one or more complete logical records.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0034'	X'0000'	USF6PEPU	PROGRAM ERROR PURGING

The remote transaction program issued an LU 6.2 SEND_ERROR verb, specifying the TYPE(PROG) parameter, and the conversation for the remote transaction program was in RECEIVE state. The LU 6.2 SEND_ERROR verb might have caused information to be purged. Purging occurs when a transaction program issues the LU 6.2 SEND_ERROR verb in RECEIVE state before receiving all the information sent by the local application, that is, all the information sent prior to the reporting of the PROGRAM_ERROR_PURGING return code to the local application. The purging can occur at the local LU, the remote LU, or both. No purging occurs when a transaction program issues the LU 6.2 SEND_ERROR verb in a CONFIRM state, or in RECEIVE state after receiving all the information sent by the local application. This RCPRI code is normally reported to the local application on an APPCCMD it issues after sending some information to the remote transaction program. However, the RCPRI code can be reported on an APPCCMD the application issues prior to sending any information, depending on the CONTROL and QUALIFY fields of the APPCCMD and when it is issued. The conversation is in RECEIVE state.

Note: This code is never reported on an APPCCMD issued on a full-duplex conversation.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0038'	X'0000'	USF6PETR	PROGRAM ERROR TRUNCATING

The remote transaction program issued an LU 6.2 SEND_ERROR verb, specifying the TYPE(PROG) parameter; the conversation for the remote transaction program was in a sending state; and the LU 6.2 SEND_ERROR verb truncated a logical record. Truncation occurs when a transaction program begins sending a logical record and then issues the LU 6.2 SEND_ERROR verb before sending the complete logical record. This return code is reported to the local application on an APPCCMD CONTROL=RECEIVE macroinstruction issued after receiving the truncated logical record. The conversation state is unchanged.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'003C'	X'0000'	USF6SENT	SERVICE ERROR NO TRUNCATION

The remote transaction program issued an LU 6.2 SEND_ERROR verb, specifying the TYPE(SVC) parameter; the conversation for the remote transaction program was in a sending state; and the LU 6.2 SEND_ERROR verb did not truncate a logical record. No truncation occurs when a transaction program issues the LU 6.2 SEND_ERROR verb before sending any logical records or after sending a complete logical record. This return code is reported to the local application on an APPCCMD CONTROL=RECEIVE macroinstruction it issues prior to receiving any logical records or after receiving one or more complete logical records. The conversation state is unchanged.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0040'	X'0000'	USF6SEPU	SERVICE ERROR PURGING

The remote transaction program issued an LU 6.2 SEND_ERROR verb, specifying the TYPE(SVC) parameter, and the conversation for the remote transaction program was in RECEIVE state. The LU 6.2 SEND_ERROR verb might have caused information to be purged. Purging occurs when a transaction program issues the LU 6.2 SEND_ERROR verb in RECEIVE state before receiving all the information sent by the local application, that is, all the information sent prior to the reporting of the SERVICE_ERROR_PURGING return code to the local application. The purging can occur at the local LU, the remote LU, or both. No purging occurs when a transaction program issues the LU 6.2 SEND_ERROR verb in a CONFIRM state, or in RECEIVE state after receiving all the information sent by the local application. This return code is normally reported to the local application on an APPCCMD it issues after sending some information to the remote transaction program. However, the return code can be reported on an APPCCMD the application issues prior to sending any information, depending on the CONTROL and QUALIFY fields of the APPCCMD and when it is issued. The conversation is in RECEIVE state.

Note: This code is never reported on an APPCCMD issued on a full-duplex conversation.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0044'	X'0000'	USF6SETR	SERVICE ERROR TRUNCATING

The remote transaction program issued an LU 6.2 SEND_ERROR verb, specifying the TYPE(SVC) parameter; the conversation for the remote transaction program was in a sending state; and the LU 6.2 SEND_ERROR verb truncated a logical record. Truncation occurs when a program begins sending a logical record and then issues the LU 6.2 SEND_ERROR verb before sending the complete logical record. This return code is reported to the local application on an APPCCMD CONTROL=RECEIVE macroinstruction issued after receiving the truncated logical record. The conversation state is unchanged.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0048'	X'0000'	USF6RFNR	RESOURCE FAILURE, NO RETRY

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. The condition is not temporary, and the application should not retry the transaction until the condition is corrected. The conversation is in END_CONV or FDX_RESET state if no log data is present. If log data is present, the conversation is in PEND_END_CONV_LOG or PEND_RESET_LOG state.

Two common failures are:

- Local LU sends unexpected control information.

For example, the conversation can be in PENDING_DEALLOCATE state, but something other than a deallocate is received, or an FMH-7 is not received when it is expected.

- Local LU sends unexpected data on the conversation.

  For example, a logical record that is not valid, PS header or FMH-7, might have been received, or a logical record is truncated by something other than an FMH-7.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'004C'	X'0000'	USF6RFRE	RESOURCE FAILURE, RETRY

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 or a modem failure. The application can retry the transaction when the error that caused the session outage has been corrected. The conversation is in END_CONV or FDX_RESET state.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0050'	X'0000'	USF6STER	STATE ERROR

The specified conversation was not in an appropriate state to issue the specified APPCCMD. For example, the application program issued APPCCMD CONTROL=SEND, QUALIFY=DATA, but the conversation was in RECEIVE state. The state of the conversation remains unchanged.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0054'	X'0000'	USF6URMD	UNRECOGNIZED MODE NAME

The APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS macroinstruction did not execute successfully because the partner LU does not recognize the specified mode name. The local and partner LUs' CNOS parameters are not changed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0058'	X'0000'	USF6UNSC	UNSUCCESSFUL, SESSION NOT AVAILABLE

The APPCCMD CONTROL=ALLOC, QUALIFY=IMMED macroinstruction issued by the local application program did not execute successfully because there was not a contention-winner session available for use by a new conversation request. This RCPRI code is returned on the unsuccessful APPCCMD.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'005C'	(all)	USF6UECR	USER ERROR CODE RECEIVED

An FMH-7 was received that contained a sense code not interpreted by VTAM. The unrecognized sense code is passed to the application program through the SENSE

field in the RPL extension. The application program must determine whether the sense code is a valid user-supplied sense code or a code that is not valid. The USER_ERROR_CODE_RECEIVED RCPRI code together with the following RCSEC subcodes (X'0000' X'0001') form the complete return code that is returned to the application. The subcode specifies whether a negative response preceded the FMH-7 containing the unrecognized sense code. The conversation is in a receiving state.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'005C'	X'0000'	USF6FNGR	FOLLOWING NEGATIVE RESPONSE

The FMH-7 containing the unrecognized sense code was received by VTAM following the receipt of a negative response.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'005C'	X'0001'	USF6WNGR	WITHOUT NEGATIVE RESPONSE

The FMH-7 containing the unrecognized sense code was not preceded by a negative response.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0060'	X'0000'	USF6NOFM	NO FMH5 AVAILABLE

The application issued an APPCCMD CONTROL=RCVFMH5, but there is currently no FMH-5 waiting to be received by the application program.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0064'	X'0000'	USF6ACFL	ACTIVATION FAILURE

An APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS macroinstruction did not execute successfully because activation for the pending active session failed. For example, the path between the application and the other LU could have been lost.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0068'	X'0000'	USF6SLEX	LU MODE SESSION LIMIT EXCEEDED

An APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS macroinstruction did not execute successfully because activating the pending active session would have caused the session limits for the mode name group to be exceeded.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'006C'	X'0000'	USF6SACT	SESSION NOT PENDING

An APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS or QUALIFY=DACTSESS macroinstruction was issued for a session that is no longer pending. The CID for the session is valid but a BIND or CINIT is no longer queued, or the session is being deactivated due to a previous error or request.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0070'	X'0000'	USF6STOR	TEMPORARY STORAGE SHORTAGE OR RESOURCE SHORTAGE

VTAM is unable to process the request because of a temporary storage shortage, a resource shortage, or other shortage.

- If a sense code is not provided, a temporary storage shortage has occurred.
- If a sense code is provided indicating insufficient resources, then a storage shortage or other resource shortage has occurred. In either of these cases, the request can be reissued (with EXECRPL, for example.) There is no state change. This return code is reported to the application program to allow time for the problem to diminish or disappear. If VTAM attempts to retry the request, the additional storage might not be available immediately, and the problem might occur again.
- If a sense code is provided other than one for insufficient resources, examine the sense code explanation to determine the action required. In this situation, whether the request can be reissued depends on the information contained in the sense code.
- If this return code is received at the completion of an APPCCMD with CONTROL=RECEIVE, OPTCD=(,XBUFLST), then a CSM buffer that meets the storage type specified in the XBUFLST-receive vector could not be obtained to receive the data, or other VTAM internal resources required to receive the data could not be obtained. The system is storage constrained. No data is received.

  The application can take several possible actions:
  - Reissue the APPCCMD several times as a temporary retry recovery action.
  - Issue a receive without the XBUFLST specification so the data can be copied into application private storage.
  - Explicitly deallocate the conversation via APPCCMD services.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0074'	X'0000'	USF6HALT	HALT ISSUED

The operator has issued a HALT command. Depending on the type of HALT, the application program can no longer issue certain macroinstructions.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0078'	X'0000'	USF6VIYA	VTAM INACTIVE FOR YOUR ACB

The association between VTAM and the application program (ACB) that was established with the OPEN macroinstruction has been broken (the ACB is in the process of being closed). This might have occurred because:

- The application program has elsewhere issued a CLOSE that has not yet completed

- VTAM has become inactive
- A VARY NET,INACT command was issued for the application program.

Any active conversations are placed in END_CONV or FDX_RESET state.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'007C'	X'0000'	USF6RQAB	REQUEST ABORTED

VTAM has rejected a request because of an error detected while processing the request or because of an error in the associated session, task, or address space. For example, an abend. An abend might or might not be retried.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0080'	X'0000'	USF6DLNR	DEALLOCATE NORMAL

The remote transaction program issued an LU 6.2 DEALLOCATE TYPE(FLUSH) verb. This return code is reported to the application program on an APPCCMD CONTROL=SEND, QUALIFY=ERROR macroinstruction issued when the conversation is in RECEIVE state. The conversation is in END_CONV state. The conversation can be in RECEIVE state or in PEND_RCV_LOG state. This return code applies only to half-duplex conversations.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0084'	X'0000'	USF6STSH	STORAGE SHORTAGE

Indicates VTAM has encountered a storage shortage when attempting to satisfy an APPCCMD CONTROL=RECEIVE or an APPCCMD CONTROL=RCVFMH5, either while storing incoming data or sending a pacing response. There is no state change.

This return code can also be issued when a storage failure occurs while processing an internal DEALLOC FLUSH request. VTAM does internal DEALLOC FLUSH processing when it receives an indication that the partner has issued an abnormal deallocation request on the full-duplex conversation.

The application should issue one of the abnormal termination APPCCMD CONTROL=DEALLOC|DEALLOCQ macroinstructions to end the conversation.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0088'	X'0000'	USF6CREJ	CANCELED BY REJECT OR ABNORMAL DEALLOCATE

The request, while in progress, was canceled by the issuance of an APPCCMD CONTROL=REJECT or abnormal deallocation APPCCMD, which has requested the termination of the current conversation and, possibly, the session.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'008C'	X'0000'	USF6PROE	PARTNER COMMITTED PROTOCOL VIOLATION

The partner LU has violated conversation protocols during the execution of this command. Notification of conversation failure will be received on a subsequent APPCCMD command. There is no state change.

Two common protocol violations are:

- Partner LU sends unexpected control information.

  For example, the conversation can be in PENDING_DEALLOCATE state, but something other than a deallocate is received, or an FMH-7 is not received when it is expected.
- Partner LU sends unexpected data on the conversation.

  For example, a logical record that is not valid, PS header or FMH-7, might have been received, or a logical record is truncated by something other than an FMH-7.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0090'	X'0000'	USF6NOTA	APPLICATION NOT APPC CAPABLE

The application program issued an APPCCMD, but the application program has APPC=NO coded on its APPL definition statement. The APPL definition statement must have APPC=YES coded before the application program can issue APPCCMD macroinstructions.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0094'	X'0000'	USF6SDRJ	INVALID CONDITION FOR SENDING DATA

This indicates that the application program issued an APPCCMD that provided data to be sent following an error on a previous QUALIFY=DATAFLU or QUALIFY=DATACON type of send (either CONTROL=SEND, CONTROL=PREPRCV or CONTROL=DEALLOC). However, data remains, held by VTAM, from the error on the previous DATAFLU or DATACON macroinstruction.

Before sending more data, issue a macroinstruction that flushes VTAM's buffers. An APPCCMD CONTROL=SEND, QUALIFY=FLUSH macroinstruction, an APPCCMD CONTROL=SEND, QUALIFY=ERROR macroinstruction, or one of the abnormal termination CONTROL=DEALLOC macroinstructions will flush the send data queue so that processing can continue.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'0098'	X'0000'	USF6STGS	TEMPORARY STORAGE SHORTAGE WHILE SENDING DATA

RCPRI,RCSEC Combinations

This indicates a temporary storage shortage has occurred while sending data. This RCPRI, RCSEC combination might be returned for one of the following macroinstructions:

- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDPROG
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDSERV
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDTIME
- APPCCMD CONTROL=DEALLOC, QUALIFY=ABNDUSER
- APPCCMD CONTROL=DEALLOCQ, QUALIFY=ABNDPROG
- APPCCMD CONTROL=DEALLOCQ, QUALIFY=ABNDSERV
- APPCCMD CONTROL=DEALLOCQ, QUALIFY=ABNDTIME
- APPCCMD CONTROL=DEALLOCQ, QUALIFY=ABNDUSER
- APPCCMD CONTROL=DEALLOC, QUALIFY=DATACON
- APPCCMD CONTROL=DEALLOC, QUALIFY=DATAFLU
- APPCCMD CONTROL=PREPRCV, QUALIFY=DATACON
- APPCCMD CONTROL=PREPRCV, QUALIFY=DATAFLU
- APPCCMD CONTROL=SEND, QUALIFY=DATA
- APPCCMD CONTROL=SEND, QUALIFY=DATACON
- APPCCMD CONTROL=SEND, QUALIFY=DATAFLU
- APPCCMD CONTROL=SEND, QUALIFY=ERROR
- APPCCMD CONTROL=SENDRCV, QUALIFY=DATAFLU.

The current position in the application-supplied data buffer (the area pointed to by the AREA field of the RPL) is returned in RPL6STBF (the current buffer) and RPL6STDS (displacement in the data). All data prior to this buffer or buffer list entry has been sent.

The user has two alternatives when this return code is received.

- Attempt to continue sending data on the conversation by issuing an APPCCMD macroinstruction with the data pointers and length set to reflect the values returned in RPL6STBF and RPL6STDS. The subsequent macroinstruction must be issued with the AREA field set with the RPL6STBF value plus the RPL6STDS value to avoid duplicating any data already sent. The data length (the RECLEN field in the RPL) must also be adjusted to indicate the amount of remaining data. Once the subsequent macroinstruction with the updated data location completes successfully, the conversation can be continued as if the storage shortage did not occur.
- Deactivate the conversation by issuing one of the abnormal termination CONTROL=DEALLOC macroinstructions, or APPCCMD CONTROL=REJECT macroinstructions. Note that REJECT must be issued to deactivate a conversation if the abnormal termination CONTROL=DEALLOC macroinstructions are unsuccessful.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'009C'	X'0001'	USF6RSTF	RESTORE REJECTED—RESTORE ISSUED BEFORE SETLOGON START

The APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE macroinstruction is issued before the SETLOGON START macroinstruction is issued.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00A0'	(all)	USF6RNAL	REQUEST NOT ALLOWED

VTAM rejected the APPCCMD because the macroinstruction request conflicts in some way with the capabilities of the session or conversation to which it applies. The REQUEST_NOT_ALLOWED RCPRI code together with one of the following RCSEC subcodes form the complete return code that is returned to the transaction program.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00A0'	X'0001'	USF6LNSE	LU PAIR DOES NOT SUPPORT SENDING EXPEDITED DATA

VTAM rejected the APPCCMD CONTROL=SENDEXPD because the negotiated support level of the current session does not support protocols needed to transmit expedited data.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00A0'	X'0002'	USF6RQBL	REQUEST BLOCKED

VTAM rejected the APPCCMD because the conversation with which it is associated is in the process of being deallocated or terminated.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00A0'	X'0003'	USF6RNEX	EXECUTION OF REQUEST TERMINATED

VTAM rejected an APPCCMD CONTROL=RCVEXPD, QUALIFY=SPEC on a half-duplex conversation because the partner LU is awaiting a change-direction or end-of-chain indicator before sending error information. No expedited information was available to be received.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00A0'	X'0004'	USF6VNVF	CONTROL/QUALIFY VALUE INVALID FOR FULL-DUPLEX CONVERSATION

VTAM rejected the APPCCMD because the CONTROL= and QUALIFY= value combination specified is not allowed for a full-duplex conversation.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00A0'	X'0005'	USF6EXRO	RSP HAS NOT BEEN RECEIVED FOR A PREVIOUS SENDEXPD REQUEST

VTAM rejected an APPCCMD CONTROL=SENDEXPD,QUALIFY=DATA or an APPCCMD CONTROL=SEND, QUALIFY=RQSEND because the response to a previously issued APPCCMD CONTROL=SENDEXPD,QUALIFY=DATA had not been received from the partner LU.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00A0'	X'0006'	USF6NAUT	PROGRAM_NOT_AUTHORIZED_FOR_ REQUESTED_FUNCTION

An application not using VTAM authorized path attempted to use the HPDT interface. The request is disallowed.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00A4'	X'0000'	USF6SPMD	MODE MUST BE RESTORED BEFORE USING

An APPCCMD macroinstruction is issued with a mode name that is pending recovery for persistent LU-LU sessions. Issue the APPCCMD CONTROL=OPRCNTL, QUALIFY=RESTORE macroinstruction to restore the mode.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00A8'	(all)	USF6ENVE	ENVIRONMENT ERROR

A macroinstruction has failed for some reason related to the system environment in which the request was processed. The RCSEC subcode identifies the specific error.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00A8'	X'0000'	USF6OSLV	OS LEVEL DOES NOT SUPPORT REQUESTED FUNCTION

A macroinstruction request required the use of an operating system service which is not supported by the active operating system level.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00A8'	X'0001'	USF6XMES	SUSPEND FAILURE

VTAM attempted to suspend processing of an APPCCMD macroinstruction issued in either cross-memory mode or in synchronous SRB-mode with OPTCD=KEEPSRB specified. The attempt failed, probably due to conditions in the operation system environment. The application may reissue the request.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00A8'	X'0002'	USF6XMER	RESUME FAILURE

VTAM attempted to resume processing of an APPCCMD macroinstruction issued in either cross-memory mode or in synchronous SRB-mode with OPTCD=KEEPSRB specified. The attempt failed. VTAM is unable to post the request complete. If the application has a LOSTERM exit, it will be scheduled with a reason code of 44. For more information about the LOSTERM exit, see *z/OS Communications Server: SNA*

Programming . The RPL is now available for reuse.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	(all)	USF6ERIN	ERROR INDICATION RECEIVED

VTAM's processing of an APPCCMD request stored on the SEND queue of a full-duplex conversation was ended because the remote transaction program or LU issued an LU 6.2 architecture verb that canceled further processing of the request. An associated Secondary Return Code value indicates the type of operation that caused the request to be ended.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0001'	USF6EIAS	DEALLOCATE ABEND PROGRAM

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because an abnormal deallocation request was issued by the remote transaction program. The FMH-7 received from the partner LU carried a sense code indicating that the remote transaction program issued a DEALLOCATE verb with TYPE(ABEND_PROG).

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0002'	USF6ERAS	DEALLOCATE ABEND SERVICE

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because an abnormal deallocation request was issued by the remote transaction program. The FMH-7 received from the partner LU carried a sense code indicating that the remote transaction program issued a DEALLOCATE verb with TYPE(ABEND_SVC).

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0003'	USF6EIAT	DEALLOCATE ABEND TIME

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because an abnormal deallocation request was issued by the remote transaction program. The FMH-7 received from the partner LU carried a sense code indicating that the remote transaction program issued a DEALLOCATE verb with TYPE(ABEND_TIMER).

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0004'	USF6EIAT	ALLOCATION ERROR

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because an abnormal deallocation request was issued by the remote transaction program. The FMH-7 received from the partner LU carried a sense code indicating that an allocation request was rejected by the remote transaction program.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0005'	USF6EIUN	UNKNOWN ERROR CODE

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because an abnormal deallocation request was issued by the remote transaction program. The FMH-7 received from the partner LU carried a sense code other than the Deallocate ABEND, Allocation Error, or Resource Failure codes. The application program must determine whether the sense code is a valid user-supplied sense code or is a code that is not valid.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0006'	USF6EIRR	RESOURCE FAILURE, RETRY

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because a failure occurred that caused the conversation to be prematurely terminated. The application can retry the transaction when the error that caused the session outage has been corrected.

RCPRI	RCSEC	ISTUSFBC EQU label	Meaning
X'00AC'	X'0007'	USF6EIRN	RESOURCE FAILURE, NO RETRY

An APPCCMD that processes on the SEND queue of a full-duplex conversation was terminated because a failure occurred that caused the conversation to be prematurely terminated. The condition is not temporary, and the application should not retry the transaction until the condition is corrected.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'(all)'	USF6NRER	NAME RESOLUTION ERROR

VTAM rejected an APPCCMD because there was an inappropriate name translation. The NAME_RESOLUTION_ERROR RCPRI code together with one of the following RCSEC subcodes form the complete return code that is returned to the transaction program.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0001'	USF6NRRE	LUNAME FOUND IN A VARIANT NAME ENTRY

VTAM rejected an APPCCMD because the LUNAME specified on the macroinstruction was found in a VARIANT_NAME entry in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0002'	USF6NRRD	NAME RETURNED DIFFERS FROM ASSOCIATED NAME

VTAM rejected an APPCCMD because the BIND RSP contained an LUNAME that is different from the associated name in the SUPPLIED_NAME entry in the LU-mode table. The association of names for the partner LU had previously occurred.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0003'	USF6NRRA	NAME RETURNED FOUND IN VARIANT_NAME ENTRY

VTAM rejected an APPCCMD because the LUNAME returned in the BIND RSP was found in a VARIANT_NAME entry in the LU-mode table. The association of names for the partner LU has not occurred.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0004'	USF6NRAP	NAME RETURNED FOUND IN SUPPLIED_NAME ENTRY

VTAM rejected an APPCCMD because the LUNAME contained in the BIND RSP was found in a SUPPLIED_NAME entry in the LU-mode table. The SUPPLIED_NAME entry was different than the entry used in the session initiation.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0005'	USF6NRNM	PARTNER NETWORK NAME MISMATCH

VTAM rejected an APPCCMD because the NETID contained in the BIND RSP was different than that previously saved in the LU-mode table for that LUNAME.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0006'	USF6NRAV	LUNAME FOUND IN AN UNUSABLE_NAME ENTRY

VTAM rejected an APPCCMD because the LUNAME specified on the macroinstruction was found in an UNUSABLE_NAME entry in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0007'	USF6NRRE	NAME RETURNED FOUND IN AN UNUSABLE_NAME ENTRY

VTAM rejected an APPCCMD because the partner LU returned an LUNAME in the BIND response that was found in an UNUSABLE_NAME entry in the LU-mode table.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B0'	X'0008'	USF6NRDN	LU NAME FOUND IN A DISASSOCIATED_NAME ENTRY

RCPRI,RCSEC Combinations

VTAM rejected an APPCCMD macroinstruction request or an operator command because the LU name specified is a DISASSOCIATED_NAME entry. This type of entry has no mode values and thus has no sessions. The LU name was previously a VARIANT_NAME entry but is no longer associated with a SUPPLIED_NAME entry.

If the request or operator command was to display information about the LU, reissue the request with a with LOGMODE=0 and any LU-specific information will be returned.

If the request was for an allocate, a CNOS must be issued to establish mode information before the allocate can be retried.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B4'	(all)	USF6CSME	CSM_DETECTED_ERROR

CSM detected an error. The CSM_DETECTED_ERROR RCPRI code together with one of the following RCSEC subcodes form the complete return code that is returned to the transaction program.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B4'	X'0001'	USF6NSPC	CSM_DETECTED_ERROR—NOT_SPECIFIED

CSM detected a problem during APPCCMD processing of the request. The specific reason for the error is not passed back to the APPCCMD application.

Upon receipt of this return code the application can:

- Optionally consider the error temporary and retry the request several times.

  Note that it is possible that the error may not recur. This temporary error condition could occur in the case where a VTAM-built parameter list to CSM is randomly corrupted on a particular request, but not on a subsequent request.
- Consider the error permanent and terminate the conversation.

Refer to *z/OS Communications Server: CSM Guide* for more information about these CSM errors.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B4'	X'0002'	USF6IBTK	CSM_DETECTED_ERROR—INVALID_BUFFER_TOKEN_SPECIFIED

The communications storage manager (CSM) detected a problem during APPCCMD processing of the request. The specific reason for the error is that CSM detected that the CSM buffer token being used for the APPCCMD is not a valid CSM buffer token.

Upon receipt of this return code the application can:

- Check the current buffer pointer (RPL6STBF) in the RPL extension to determine the address of the buffer list entry that was processed when the error occurred.
- Optionally consider the error temporary and retry the request several times.

Note that it is possible that the error may not recur. This temporary error condition could occur in the case where a VTAM-built parameter list to CSM is randomly corrupted on a particular request, but not on a subsequent request.

- Consider the error permanent and terminate the conversation.
- Continue using the conversation with a different CSM buffer.

Refer to*z/OS Communications Server: CSM Guide* for more information about these CSM errors.

RCPRI	RCSEC	ISTUSFBC EQU Label	Meaning
X'00B4'	X'0003'	USF6IIID	CSM_DETECTED_ERROR—INVALID_INSTANCE_ID_SPECIFIED

The communications storage manager (CSM) detected a problem during APPCCMD processing of the request. The specific reason for the error is that CSM detected that the instance ID portion of the CSM buffer token being used for the APPCCMD is not a valid CSM instance ID. Since the instance ID is not valid, it is possible that the CSM buffer being specified on the APPCCMD has been previously freed and a new instance ID has been assigned to the storage by CSM.

Upon receipt of this return code the application can:

- Check the current buffer pointer (RPL6STBF) in the RPL extension to determine the address of the buffer list entry that was processed when the error occurred.
- Optionally consider the error temporary and retry the request several times.

  Note that it is possible that the error may not recur. This temporary error condition could occur in the case where a VTAM-built parameter list to CSM is randomly corrupted on a particular request, but not on a subsequent request.
- Consider the error permanent and terminate the conversation.
- Continue using the conversation with a different CSM buffer.

Refer to *z/OS Communications Server: CSM Guide* for more information about these CSM errors.

RTNCD, FDB2 Information for LU 6.2

While most of the LU 6.2 feedback information from errors is found in the RCPRI and RCSEC fields, some error return codes in the RPL RTNCD and FDB2 fields are meaningful for LU 6.2 applications. The X'00', X'0B' combination in the RPL indicates some problem might have occurred while the macroinstruction was executing. RCPRI and RCSEC should be used for further diagnosis. The other RTNCD, FDB2 combinations refer to attempts to start an LU 6.2 session independent of VTAM or attempts to use non-APPCCMD macroinstructions for APPCCMD functions. Following are the relevant codes.

RTNCD	FDB2	ISTUSFBC EQU Label	Meaning
X'00'	X'0B'	USF6APPC	CONDITIONAL COMPLETION FOR APPCCMD

Some type of error might have occurred on an APPCCMD macroinstruction. For further problem determination, refer to the primary and secondary return codes in the RPL extension. These fields are RPL6RCPR and RPL6RCSC.

RTNCD, FDB2 Combinations

RTNCD	FDB2	ISTUSFBC EQU Label	Meaning
X'04'	X'05'	USFNQN	SYMBOLIC NAME KNOWN BY NETWORK-QUALIFIED NAME ONLY

A real-to-symbolic translation request is made, and NIBNET is filled in with a network identifier, but VTAM cannot provide a symbolic name. VTAM knows this resource only by its network-qualified name. No symbolic name represents this resource. Do one of the following:

- Use the network-qualified name
- Define a symbolic name to represent this resource

RTNCD	FDB2	ISTUSFBC EQU Label	Meaning
X'10'	X'13'	USF6APRJ	ATTEMPT TO START 6.2 SESSION: REQUEST REJECTED

An LU 6.2 application program has tried to start an LU 6.2 session independent of VTAM. No pending sessions have been disturbed. This occurs when an OPNDST is issued with an LU 6.2 user-specified BIND.

RTNCD	FDB2	ISTUSFBC EQU Label	Meaning
X'10'	X'14'	USF6APST	ATTEMPT TO START 6.2 SESSION: PENDING SESSION TERMINATED

An LU 6.2 application program has tried to start an LU 6.2 session independent of VTAM. The pending session has been terminated. This occurs when the LOGMODE specified on an OPNDST resolves to an LU 6.2 BIND or when OPNSEC is issued for an LU 6.2 BIND.

RTNCD	FDB2	ISTUSFBC EQU Label	Meaning
X'10'	X'15'	USF6APIS	AN APPCCMD MUST BE ISSUED

An OPNDST or CLSDST has been issued for a pending LU 6.2 session. An APPCCMD CONTROL=OPRCNTL, QUALIFY=ACTSESS or QUALIFY=DACTSESS macroinstruction must be issued for this session.

RTNCD	FDB2	ISTUSFBC EQU Label	Meaning
X'14'	X'7F'	USF6PENA	POLICING ERROR — NON-APPC MACRO

An application program issued a non-APPCCMD macroinstruction to establish an LU 6.2 session, or issued a non-APPCCMD macroinstruction against a current LU 6.2 session.

Chapter 3. DSECTs

This chapter contains the LU 6.2 DSECTs. For general information on the use and purpose of DSECTs, refer to *z/OS Communications Server: SNA Programming*.

The DSECTs are shown as an abbreviated form of an assembler listing. The first column indicates the offsets within the DSECT and is the “LOC” column of an assembler listing. The object code, address columns and statement number columns of the listing, however, are not included. The source statements and comments are found next to the offset column. All numbers in the offset column are in hexadecimal.

BIND Image (ISTDBIND)

```
  LOC     SOURCE STATEMENT
000000 ISTDBIND DSECT
000000 BINFMTY  DS    C                   BIND FORMAT AND TYPE
       BINFMT   EQU   X'F0'               BIND FORMAT
       *    VALUES FOR BINFMT (FORMAT)
       BINFMT0  EQU   X'00'               FORMAT 0
       BINTYPE  EQU   X'0F'               BIND TYPE
       *    VALUES FOR BINTYPE (TYPE)
       BINNEGO  EQU   X'00'               NEGOTIABLE
       BINONEGO EQU   X'01'               NON NEGOTIABLE
       BINCOLD  EQU   X'01'               NON NEGOTIABLE
000001 BINFM    DS    C                   FUNCTION MANAGEMENT
       *                                  PROFILE
       *    VALUES FOR BINFM - FUNCTION MANAGEMENT PROFILE
       BINFM19  EQU   X'13'               FM PROFILE 19
000002 BINTS    DS    C                   TRANSMISSION SERVICES
       *                                  PROFILE
       *    VALUES FOR BINTS (TRANSMISSION SERVICES PROFILE)
       BINTS7   EQU   X'07'               SEQ NOS - NO RESET STATE
       BINTS4   EQU   X'04'               SEQ NOS - RESET STATE
       BINTS3   EQU   X'03'               SEQ NOS - RESET STATE
       BINTS2   EQU   X'02'               SEQ NOS - NO RESET STATE
       BINTS1   EQU   X'01'               NOT VALID ON LU-LU SESSION
       BINTS0   EQU   X'00'               NOT VALID ON LU-LU SESSION
000003 BINPRIP  DS    C                   PRIMARY LU PROTOCOLS FOR
       *                                  SENDING FM DATA
       BINPCHN  EQU   X'80'               1 = MULTIPLE RU CHAINS
       *                                  0 = SINGLE RU CHAINS
       BINPMCH  EQU   X'40'               1 = MULTIPLE OUTSTANDING
       *                                      CHAINS (DELAYED
       *                                      REQUEST MODE)
       *                                  0 = SINGLE OUTSTANDING
       *                                      CHAIN (IMMEDIATE
       *                                      REQUEST MODE)
       BINPCHNR EQU   X'30'               CHAIN RESPONSE PROTOCOL(SEE
       *                                  BINSCHNR BELOW FOR VALUES)
       BINRSV01 EQU   X'0C'               RESERVED
       BINPCMP  EQU   X'02'               1 = COMPRESSION MAY BE
       *                                      USED
       *                                  0 = COMPRESSION MUST NOT
       *                                      BE USED
       BINPSEB  EQU   X'01'               1 = PRIMARY MAY SEND EB
       *                                  0 = PRIMARY WILL NOT
       *                                      SEND EB
000004 BINSECP  DS    C                   SECONDARY LU PROTOCOLS FOR
       *                                  SENDING FM DATA
       BINSCHN  EQU   X'80'               1 = MULTIPLE RU CHAINS
       *                                  0 = SINGLE RU CHAIN
       BINSMCH  EQU   X'40'               1 = MULTIPLE OUTSTANDING
```

© Copyright IBM Corp. 1988, 2001

```
        *                                   CHAINS (DELAYED
        *                                   REQUEST MODE)
        *                               0 = SINGLE OUTSTANDING
        *                                   CHAIN (IMMEDIATE
        *                                   REQUEST MODE)
        BINSCHNR EQU   X'30'            CHAIN RESPONSE PROTOCOLS
        ***********************************************************************
        *    VALUES FOR BINPCHNR/BINSCHNR (TYPE OF RESPONSES ASKED
        *    FOR BY REQUESTS FROM PRIMARY/SECONDARY)
        ***********************************************************************
        BINNYRSP EQU   X'30'            DEFINITE OR EXCEPTION
        *                               RESPONSE
        BINDFRSP EQU   X'20'            DEFINITE RESPONSE
        BINEXRSP EQU   X'10'            EXCEPTION RESPONSE
        BINNORSP EQU   X'00'            NO RESPONSE
        BINRSV02 EQU   X'0C'            RESERVED
        BINSCMP  EQU   X'02'            1 = COMPRESSION MAY BE
        *                                   USED
        *                               0 = COMPRESSION MUST NOT
        *                                   BE USED
        BINSSEB  EQU   X'01'            1 = SECONDARY MAY SEND EB
        *                               0 = SECONDARY WILL NOT
        *                                   SEND EB
000005  BINCMNP  DS    C                COMMON LU PROTOCOLS
        BINWBREQ EQU   X'80'            WHOLE-BINS-REQUIRED INDICATOR
        *
        BINFMHD  EQU   X'40'            1 = FM HEADERS MAY
        *                                   BE USED
        *                               0 = FM HEADERS MUST NOT
        *                                   BE USED
        BINBRAK  EQU   X'20'            1 = BRACKETS WILL BE USED
        *                                   AND RESET STATE IS
        *                                   BETWEEN-BRACKETS
        *                               0 = BRACKETS WILL NOT
        *                                   BE USED OR,IF USED,
        *                                   RESET STATE IS IN-
        *                                   BRACKETS
        BINBKTR  EQU   X'10'            1 = CONDITIONAL BRACKETS
        *                                   TERMINATION
        *                               0 = UNCONDITIONAL BRACKETS
        *                                   TERMINATION
        BINALT   EQU   X'08'            1 = ALTERNATE CODE MAY
        *                                   BE USED
        *                               0 = ALTERNATE CODE MUST
        *                                   NOT BE USED
        BINRSV04 EQU   X'06'            RESERVED
        BINQUE   EQU   X'01'            BIND-QUEUEING INDICATOR
000006  BINCMNP2 DS    C                COMMON LU PROTOCOLS
        BINFMTRM EQU   X'C0'            SEND/RECEIVE MODE
        *    VALUES FOR BINFMTRM
        BINMSTSL EQU   X'C0'            RESERVED
        BINHDXFF EQU   X'80'            HDX FLIP FLOP
        BINHDXC  EQU   X'40'            HDX CONTENTION
        BINFLDPX EQU   X'00'            FULL DUPLEX
        BINRCVR  EQU   X'20'            1 = SYMMETRIC
        *                                   RESPONSIBILITY FOR
        *                                   RECOVERY
        *                               0 = CONTENTION LOSER (SEE
        *                                   BINBKFS BELOW)
        *                                   RESPONSIBLE FOR
        *                                   RECOVERY
        BINBKFS  EQU   X'10'            1 = PRIMARY IS BRACKETS
        *                                   FIRST SPEAKER AND CONTENTION
        *                                   WINNER; SECONDARY IS BRACKETS
        *                                   BIDDER AND CONTENTION LOSER
        *                               0 = SECONDARY IS BRACKETS
        *                                   FIRST SPEAKER AND CONTENTION
```

```
       *                                    WINNER; PRIMARY IS BRACKETS
       *                                    BIDDER AND CONTENTION LOSER
       BINASCC  EQU   X'0C'               ALTERNATE CODE PROCESSING IDENTIFIER
       *
       *                                  00=ASCII7
       *                                  01=ASCII8
       BINCTLV  EQU   X'02'               CONTROL VECTORS ARE INCLUDED
       *                                  AFTER THE SLU NAME
       BINCONR  EQU   X'01'               RESET STATE FOR HDX FLIP-FLOP
       *                                  (E.G. AT START OF SESSION)
       *                                  1 = PRIMARY SENDS FIRST
       *                                      WHEN DATA TRAFFIC
       *                                      RESET STATE IS LEFT
       *                                  0 = SECONDARY SENDS FIRST
000007 BINTSU   DS    CL6                 TS USAGE
00000D BINPRSVC DS    CL12                PRESENTATION SERVICES
000019 BINCRCTL DS    CL1                 CRYPTOGRAPHY CONTROL BYTE
       *              VALUES FOR BINCRCTL
       BINNOCRY EQU   X'00'               NO CRYPTOGRAPHY
       BINCRYCA EQU   X'09'               CAPABLE OF CRYPTOGRAPHY
       BINCRYSL EQU   X'19'               SELECTIVE CRYPTOGRAPHY
       BINCRYRQ EQU   X'39'               REQUIRED CRYPTOGRAPHY
       *
       BINCEUMP EQU   X'C0'               EU/PRIVATE CRYPTOGRAPHY FLAGS
       *              VALUES FOR BINCEUMP
       BINCEUPS EQU   X'80'               SYSTEM KEY, PRIVATE PROTOCOL
       BINCEUPP EQU   X'40'               PRIVATE KEY, PRIVATE PROTOCOL
       BINCEUNP EQU   X'00'               NO PRIVATE/EU PROTOCOL
       *
       BINCSESS EQU   X'30'               SESSION LEVEL CRYPTOGRAPHY FLAGS
       *              VALUES FOR BINCSESS
       BINCSENP EQU   X'00'               NO CRYPTOGRAPHY
       BINCSESP EQU   X'10'               SELECTIVE CRYPTOGRAPHY
       BINCSESR EQU   X'30'               REQUIRED CRYPTOGRAPHY
       *
       BINCLEN  EQU   X'0F'               LENGTH OF CRYPTOGRAPHY FIELD
       *
       *
00001A BINPRIML DS    C                   PRIMARY LU NAME LENGTH
00001B BINPRIMN DS    CL8                 PRIMARY LU NAME
       *
       *  INCLUDE FOR COMPATIBILITY
       *
000023         ORG   BINPRIMN
00001B BINPRIM  DS    8C                  PRIMARY LU NAME
       *
000023 BINUSEL  DS    C                   USER DATA LENGTH
       BINUSE   EQU   *                   USER DATA
       BINUSERD EQU   X'00'               USER DATA LENGTH DEFAULT
       *
       *  OVERLAY FOR 'BINTSU' (TS USAGE)
000024         ORG   BINTSU
000007 BINAPACE DS    C                   SLU SEND PACING
       BINSP2ST EQU   X'80'               NUMBER OF PACING STAGES FROM
       *                                  SLU TO PLU ( NOTE-REVERSE OF
       *                                  BINPS1ST)
       *                                  1 = TWO STAGES
       *                                  0 = ONE STAGE
       BINRSV43 EQU   X'40'               RESERVED
       BINAPACM EQU   X'3F'               SLU SEND PACING COUNT
000008 BINRPACE DS    C                   SLU RECEIVE PACING
       BINASPI  EQU   X'80'               ADAPTIVE SESSION PACING INDICATOR
       *
       BINRSV07 EQU   X'40'               RESERVED
       BINRPACM EQU   X'3F'               SLU RECEIVE PACING COUNT
000009 BINRUSZ  DS    0CL2                RU SIZES
000009 BINSRUSZ DS    C                   SLU MAXIMUM SEND RU SIZE
```

```
       BINSRUSS EQU   X'80'               RU SIZE IS SPECIFIED
00000A BINPRUSZ DS    C                   PLU MAXIMUM SEND RU SIZE
       BINPRUSS EQU   X'80'               RU SIZE IS SPECIFIED
       *  VALUES FOR BINSRUSZ AND BINPRUSZ (RU SIZES) EXCEPT RU SIZE
       *  SPECIFIED
       BINRU256 EQU   X'85'               256 BYTE RU (8*2**5)
       BINR4096 EQU   X'89'               4096 BYTE RU (8*2**9)
       BIN61440 EQU   X'FC'               61440 BYTE RU (15*2**12)
       BINRU1K  EQU   X'87'               1024 BYTE RU (8*2**7)
       BINRUSZM EQU   X'F0'               MANTISSA (M)
       BINRUSZE EQU   X'0F'               EXPONENT (E)  SIZE=M*2**E
00000B BINSPACE DS    C                   PLU SEND PACING
       BINPS1ST EQU   X'80'               NUMBER OF PACING STAGES FROM
       *                                  PLU TO SLU (NOTE-REVERSE OF
       *                                  BINSP2ST)
       *                                  1 = ONE STAGE
       *                                  0 = TWO STAGE
       BINRSV44 EQU   X'40'               RESERVED
       BINSPACM EQU   X'3F'               PLU SEND PACING COUNT
00000C BINBPACE DS    C                   PLU RECEIVE PACING
       BINRSV10 EQU   X'C0'               RESERVED
       BINBPACM EQU   X'3F'               PLU RECEIVE PACING COUNT
       *
       ***********************************************************************
       *  OVERLAY FOR 'BINPRSVC' (PRESENTATION SERVICES)
       ***********************************************************************
00000D         ORG   BINPRSVC
00000D BINLUP   DS    C                   PS PROFILE
       *  VALUES FOR BINLUP (PS PROFILE)
       BINPSFMT EQU   X'80'               PS USAGE FIELD FORMAT
       BINLUTYP EQU   X'7F'               LU TYPE
       BINLUP6C EQU   X'06'               LU TYPE 6
       BINLUP4C EQU   X'04'               LU TYPE 4
       BINLUP3C EQU   X'03'               LU TYPE 3
       BINLUP2C EQU   X'02'               LU TYPE 2
       BINLUP1C EQU   X'01'               LU TYPE 1
       BINLUP0C EQU   X'00'               LU TYPE 0
       *
00000E BINPSCHR DS    CL11                PS PROFILE DEPENDENT
       *                                  PRESENTATION SERVICES
       *
       ***********************************************************************
       *  OVERLAY FOR 'BINPSCHR' (PRESENTATION SERVICES CHARACTERISTICS
       *                          FOR PS PROFILE 1)
       ***********************************************************************
000019         ORG   BINPSCHR
00000E BINLUP1  DS    X                   PS PROFILE 1 FMHS AND DSP
       BINFMHS1 EQU   X'F0'               FM HEADER SUBSET
       *  VALUES FOR BINFMHS1
       BINFMS3C EQU   X'30'               DATA MANAGEMENT SUBSET
       BINFMS2C EQU   X'20'               TYPE 1 HEADERS
       BINFMS1C EQU   X'10'               TYPE 1 HEADERS WITH
       *                                  RESTRICTIONS
       BINFMS0C EQU   X'00'               NO FM HEADERS ALLOWED
       BINDSP1  EQU   X'0F'               DATA STREAM PROFILE
       *  VALUES FOR BINDSP1 (DATA STREAM PROFILE)
       BINDSP1C EQU   X'01'               BASIC CONTROLS, CARDS MAY
       *                                  SPAN RUS
       BINDSP0C EQU   X'00'               BASIC CONTROLS
00000F BINPLUS1 DS    0XL5                PLU USAGE
00000F BINPFMF1 DS    0XL2                FMH SUBSET DEPENDENT
       *                                  FLAGS
00000F BINPFMB1 DS    X                   FIRST BYTE
000010 BINPFMB2 DS    X                   SECOND BYTE
000011 BINPDSP1 DS    0XL2                DATA STREAM FLAGS FOR
       *                                  DSP0 AND DSP1
000011 BINPDSB1 DS    X                   FIRST BYTE
```

```
000012 BINPDSB2 DS    X                   SECOND BYTE
000013 BINPMED1 DS    X                   MEDIA FLAGS
000014 BINSLUS1 DS    0XL5                SLU USAGE
000014 BINSFMF1 DS    0XL2                FMH SUBSET DEPENDENT
       *                                  FLAGS
000014 BINSFMB1 DS    X                   FIRST BYTE
000015 BINSFMB2 DS    X                   SECOND BYTE
000016 BINSDSP1 DS    0XL2                DATA STREAM FLAGS FOR
       *                                  DSP0 AND DSP1
000016 BINSDSB1 DS    X                   FIRST BYTE
000017 BINSDSB2 DS    X                   SECOND BYTE (RESERVED)
000018 BINSMED1 DS    X                   MEDIA FLAGS
       *
       *  FLAGS FOR LU PROFILE 1
       *
       *  FLAGS FOR BINPFMB1 AND BINSFMB1 (FIRST BYTE OF FM
       *                                   HEADER FLAGS)
       BINDESTS EQU   X'80'               0 = TWO DESTINATIONS MAY
       *                                      BE OUTSTANDING
       *                                  1 = THREE DESTINATIONS MAY
       *                                      BE OUTSTANDING
       BINCMPCT EQU   X'40'               0 = WILL NOT SEND COMPACTION
       *                                      TABLE/WILL NOT BE QUERIED
       *                                      FOR COMPACTION TABLES
       *                                  1 = MAY SEND COMPACTION
       *                                      TABLE/MAY BE QUERIED FOR
       *                                      COMPACTION TABLES
       BINPDIR  EQU   X'20'               0 = PDIR WILL NOT BE SENT
       *                                  1 = PDIR MAY BE SENT
       BINRSV09 EQU   X'1F'               RESERVED FOR FMHS1
       *  ADDITIONAL FLAGS FOR FMHS3
       BINKDDSI EQU   X'10'               0 = KEYED DIRECT DATA SET
       *                                      WILL NOT BE SENT
       *                                  1 = KEYED DIRECT DATA SET
       *                                      MAY BE SENT
       BINSDSI  EQU   X'08'               0 = SEQUENTIAL DATA SETS
       *                                      WILL NOT BE SENT
       *                                  1 = SEQUENTIAL DATA SETS
       *                                      MAY BE SENT
       BINSAI   EQU   X'04'               0 = SEQUENTIAL ACCESS TO
       *                                      ADDRESSED DIRECT DATA
       *                                      SET WILL NOT BE SENT
       *                                  1 = SEQUENTIAL ACCESS TO
       *                                      ADDRESSED DIRECT DATA
       *                                      SET MAY BE SENT
       BINSIDS  EQU   X'02'               0 = SERIES ID NOT
       *                                      SUPPORTED (WITH STATUS
       *                                      IN REPLY)
       *                                  1 = SERIES ID SUPPORTED
       *                                      (WITH STATUS IN REPLY)
       BINARRR  EQU   X'01'               0 = ADD REPLICATE,
       *                                      REPLACE REPLICATE NOT
       *                                      SUPPORTED
       *                                  1 = ADD REPLICATE,
       *                                      REPLACE REPLICATE
       *                                      SUPPORTED
       *  FLAGS FOR BINPFMB2 AND BINSFMB2 (SECOND BYTE OF FM HEADER FLAGS)
       BINRSV17 EQU   X'FF'               RESERVED FOR FMHS1
       *  ADDITIONAL FLAGS FOR FMHS3
       BINRSV16 EQU   X'80'               RESERVED
       BINQDSI  EQU   X'40'               0 = QUERY FOR DESTINATION
       *                                      SELECTION NOT SUPPORTED
       *                                  1 = QUERY FOR DESTINATION
       *                                      SELECTION SUPPORTED
       BINCSDS  EQU   X'20'               0 = CREATE / SCRATCH /
       *                                      SCRATCH ALL DATA
       *                                      SET NOT ALLOWED
```

```
*                                   1 = CREATE / SCRATCH /
*                                       SCRATCH ALL DATA
*                                       SET NOT ALLOWED
BINXFPD  EQU   X'10'                0 = EXECUTE PROGRAM OFFLINE
*                                       NOT ALLOWED
*                                   1 = EXECUTE PROGRAM OFFLINE
*                                       ALLOWED
BINRSV11 EQU   X'0F'                RESERVED FOR FMHS3
*
*  FLAGS FOR 'BINPDSB1 AND BINSDSB1' (PLU/SLU DATA STREAM
*                                     FLAGS FOR DSP0 AND DSP1)
*  NL AND FF MAY BE SENT IN ANY SUBSET. EACH SUBSET BELOW CONTAINS
*  EVERY PRECEDING SUBSET (E.G. IF AN LU CAN SEND THE HORIZONTAL
*  FORMAT SUBSET, IT CAN ALSO SEND THE FULL BASE SET)
BININTR  EQU   X'80'                0 = FULL BASE SET DATA
*                                       STREAM (BS,CR,LF,ENP,
*                                       INP,HT,VT) WILL NOT
*                                       BE SENT
*                                   1 = FULL BASE SET DATA
*                                       STREAM (BS,CR,LF,ENP,
*                                       INP,HT,VT) MAY BE
*                                       SENT
BINHFDS  EQU   X'40'                0 = HORIZONTAL FORMAT,
*                                       DATA STREAM(SHF) WILL
*                                       NOT BE SENT
*                                   1 = HORIZONTAL FORMAT,
*                                       DATA STREAM(SHF) MAY
*                                       BE SENT
BINVTDS  EQU   X'20'                0 = VERTICAL FORMAT
*                                       DATA STREAM (SVF)
*                                       WILL NOT BE SENT
*                                   1 = VERTICAL FORMAT
*                                       DATA STREAM (SVF)
*                                       MAY BE SENT
BINVSDS  EQU   X'10'                0 = VERTICAL CHANNEL DATA
*                                       STREAM (SVF(CHANNELS),SCF,
*                                       SEL) WILL NOT BE
*                                       SENT
*                                   1 = VERTICAL CHANNEL DATA
*                                       STREAM (SVF(CHANNELS),SCF,
*                                       SEL) MAY BE SENT
BINSLD   EQU   X'08'                0 = SLD WILL NOT BE SENT
*                                   1 = SLD MAY BE SENT
BINRSV40 EQU   X'06'                RESERVED
BINTRNDS EQU   X'01'                0 = TRANSPARENCY DATA
*                                       STREAM (TRN,IRS) WILL
*                                       NOT BE SENT
*                                   1 = TRANSPARENCY DATA
*                                       STREAM (TRN,IRS)
*                                       MAY BE SENT
*  FLAGS FOR BINPDSB2
*
BINUAINT EQU   X'80'                0 = SLU WILL INITIATE
*                                       ATTENDED
*                                   1 = SLU WILL INITIATE
*                                       UNATTENDED
BINUAALT EQU   X'40'                0 = DURING SESSION SLU
*                                       WILL NOT ALTERNATE
*                                       BETWEEN ATTENDED AND
*                                       UNATTENDED
*                                   1 = DURING SESSION SLU
*                                       WILL ALTERNATE
*                                       BETWEEN ATTENDED AND
*                                       UNATTENDED
BINRSV41 EQU   X'3F'                RESERVED
*
*  FLAGS FOR BINPMED1 AND BINSMED1 (PLU/SLU MEDIA FLAGS)
```

```
       BINDOCMT EQU   X'80'               0 = DOCUMENT FORMAT WILL
       *                                      NOT BE SENT
       *                                  1 = DOCUMENT FORMAT
       *                                      MAY BE SENT
       BINCARD  EQU   X'40'               0 = CARD FORMAT WILL NOT
       *                                      BE SENT
       *                                  1 = CARD FORMAT MAY BE
       *                                      SENT
       BINXCHNG EQU   X'20'               0 = EXCHANGE MEDIA FORMAT
       *                                      WILL NOT BE SENT
       *                                  1 = EXCHANGE MEDIA FORMAT
       *                                      MAY BE SENT
       BINDISK  EQU   X'10'               0 = DISK FORMAT WILL NOT
       *                                      BE SENT
       *                                  1 = DISK FORMAT MAY BE
       *                                      SENT
       BINXCDF  EQU   X'08'               0 = EXTENDED CARD FORMAT
       *                                      WILL NOT BE SENT
       *                                  1 = EXTENDED CARD FORMAT
       *                                      MAY BE SENT
       BINXDOCF EQU   X'04'               0 = EXTENDED DOCUMENT
       *                                      FORMAT WILL NOT BE
       *                                      SENT
       *                                  1 = EXTENDED DOCUMENT
       *                                      FORMAT MAY BE SENT
       BINCDEDS EQU   X'02'               0 = SLU MAY SEND CD
       *                                      EVERY EDS
       *                                  1 = SLU MUST SEND CD
       *                                      EVERY EDS
       *                                      (THIS FLAG APPLIES
       *                                       TO BINPMED1)
       BIN1CMP1 EQU   X'02'               APPLIES ONLY to BINSMED1
       *                                  (SEE BINCMP1 AND BINCMP2)
       BIN1CMP2 EQU   X'01'               (SEE BINCMP1 AND BINCMP2)
       *
       ***********************************************************************
       *  OVERLAY FOR 'BINPSCHR' (PRESENTATION SERVICES CHARACTERISTICS
       *                          FOR PS PROFILE 2)
       ***********************************************************************
000019         ORG   BINPSCHR
00000E BINDFLAG DS    XL1                 DEVICE FLAG
       BINSEDS  EQU   X'80'               EXTENDED 3270 DATA STREAM
00000F BINRSV14 DS    XL4                 RESERVED
000013 BINSCRSZ DS    0XL5                PRESENTATION SPACE SIZE
000013 BINSPRIR DS    FL1                 PRIMARY (DEFAULT) NUMBER
       *                                  OF ROWS
000014 BINSPRIC DS    FL1                 PRIMARY (DEFAULT) NUMBER
       *                                  OF COLUMNS
000015 BINSALTR DS    FL1                 ALTERNATE NUMBER OF ROWS
000016 BINSALTC DS    FL1                 ALTERNATE NUMBER OF COLUMNS
000017 BINPRESZ DS    FL1                 PRESENTATION SPACE SIZE
       *  VALUES FOR BINPRESZ (PRESENTATION SPACE SIZE)
       BINPSZRC EQU   X'7F'               PRESENTATION SPACE HAS
       *                                  DEFAULT AND ALTERNATE
       *                                  SIZES AS DEFINED IN
       *                                  DEFAULT, ALTERNATE
       *                                  ROW/COL FIELDS
       BINPSFX  EQU   X'7E'               PRESENTATION SPACE IS
       *                                  FIXED SIZE AS DEFINED BY
       *                                  ROW/COL VALUES IN
       *                                  DEFAULT ROW/COL FIELDS
       BINPSZ3  EQU   X'03'               24X80 DEFAULT UNDEFINED ALTERNATE
       *                                  DO WRITE STRUCTURED FIELD QUERY
       *                                  TO IDENTIFY ALTERNATE
       BINPSZ2  EQU   X'02'               24X80 ROW X COLUMN
       BINPSZ1  EQU   X'01'               12X40 ROW X COLUMN
       BINPSZ0  EQU   X'00'               UNDEFINED ROW X COLUMN
```

```
000018 BIN2COMP DS    X                   COMPRESSION FLAGS
       BIN2CMP1 EQU   X'02'               APPLIES ONLY to BINSMED1
       *                                  (SEE BINCMP1 AND BINCMP2)
       BIN2CMP2 EQU   X'01'               (SEE BINCMP1 AND BINCMP2)
       *
       ********************************************************************
       *  OVERLAY FOR 'BINPSCHR' (PRESENTATION SERVICES CHARACTERISTICS
       *                          FOR PS PROFILE 3)
       ********************************************************************
000019         ORG   BINPSCHR
00000E BINRSV26 DS    XL5                 RESERVED
000013 BINBFRSZ DS    0XL4                PRESENTATION SPACE SIZE
000013 BINBFRDR DS    FL1                 PRIMARY (DEFAULT) NUMBER OF ROWS
000014 BINBFRDC DS    FL1                 PRIMARY (DEFAULT) NUMBER OF
       *                                  COLUMNS
000015 BINBFRAR DS    FL1                 ALTERNATE NUMBER OF ROWS
000016 BINBFRAC DS    FL1                 ALTERNATE NUMBER OF COLUMNS
000017 BINBDESC DS    FL1                 PRESENTATION SPACE
       *                                  SIZE SPECIFICATION:
       *                                       0 = MAXIMUM
       *                                       1 = 480 CHAR
       *                                       2 = 1920 CHAR
       *                                  X'7E' = FIXED SIZE(SEE
       *                                          BINBFRDR AND
       *                                          BINBFRDC)
       *                                  X'7F' = VARIABLE SIZE AS
       *                                          DEFINED BY
       *                                          BINBFRSZ
       BINBFSIZ EQU   X'7F'               SEE ABOVE
       BINBFSZF EQU   X'7E'               SEE ABOVE
000018 BIN3COMP DS    X                   COMPRESSION FLAGS
       BIN3CMP1 EQU   X'02'               APPLIES ONLY to BINSMED1
       *                                  (SEE BINCMP1 AND BINCMP2)
       BIN3CMP2 EQU   X'01'               (SEE BINCMP1 AND BINCMP2)
       *
       ********************************************************************
       *  OVERLAY FOR 'BINPSCHR' (PRESENTATION SERVICES CHARACTERISTICS
       *                          FOR PS PROFILE 4)
       ********************************************************************
000019         ORG   BINPSCHR
00000E BINPSNDO DS    0XL4                PLU SEND CAPABILITY
00000E BINPDSPP DS    X                   PRINTER DATA STREAM
       *                                  PROFILE
       BINPBDSP EQU   X'80'               BASE DATA STREAM PROFILE
       *                                  0 = NOT SUPPORTED
       *                                  1 = SUPPORTED
       BINRSV46 EQU   X'40'               RESERVED
       BINPJOB  EQU   X'20'               JOB SCS SUBSET
       *                                  0 = NOT SUPPORTED
       *                                  1 = SUPPORTED
       BINRSV47 EQU   X'10'               RESERVED
       BINWPRAW EQU   X'08'               WORD PROCESSING RAW FORM
       *                                  0 = NOT SUPPORTED
       *                                  1 = SUPPORTED
       BINRSV48 EQU   X'07'               RESERVED
00000F BINADSPP DS    X                   ADDITIONAL DATA STREAM
       *                                  PROFILE
       BINRSV49 EQU   X'80'               RESERVED
       BINADSCD EQU   X'40'               0 = CARD NOT SUPPORTED
       *                                  1 = CARD SUPPORTED
       BINRSV29 EQU   X'3F'               RESERVED
000010 BINCSLP  DS    X                   CONSOLE
       BINCBDSP EQU   X'80'               BASE DATA STREAM PROFILE
       *                                  0 = NOT SUPPORTED
       *                                  1 = SUPPORTED
       BINRSV50 EQU   X'40'               RESERVED
       BINCJOB  EQU   X'20'               JOB SCS SUBSET
```

```
       *                                0 = NOT SUPPORTED
       *                                1 = SUPPORTED
       BINRSV51 EQU   X'1F'             RESERVED
000011 BINFMHUP DS    X                 FM/FMH USAGE
       BINSSDAT EQU   X'80'             RESERVED
       BINDSSTO EQU   X'60'             00= 1 LEVEL DESTINATION
       *                                    SELECTION SUSPENSION
       *                                    STACK
       *                                01= 2 LEVEL DESTINATION
       *                                    SELECTION SUSPENSION
       *                                    STACK
       *                                10= RESERVED
       *                                11= 3 LEVEL DESTINATION
       *                                    SELECTION SUSPENSION
       *                                    STACK
       BINRSV52 EQU   X'1E'             RESERVED
       BINKIXS  EQU   X'01'             0 = SLU NEED NOT RECEIVE
       *                                    CD ON EVERY EDS
       *                                1 = SLU MUST RECEIVE CD
       *                                    ON EVERY EDS
000012 BINSSNDO DS    0XL4              SLU SEND CAPABILITY
000012 BINPDSPS DS    X                 PRINTER DATA STREAM
       *                                PROFILE (SEE BINPDSPP)
000013 BINADSPS DS    X                 ADDITIONAL DATA STREAM
       *                                PROFILE (SEE BINADSPP)
000014 BINCSLS  DS    X                 CONSOLE (SEE BINCSLP)
000015 BINFMHUS DS    X                 FM/FMH USAGE (SEE
       *                                BINFMHUP; MEANING FOR
       *                                BINKIXS IS: 0 = PLU NEED
       *                                NOT RECEIVE CD ON EVERY
       *                                EDS, 1 = PLU MUST RECEIVE
       *                                CD ON EVERY EDS)
000016 BINCSO   DS    X                 CODE SELECTION
       BINCSOR  EQU   X'F0'             REPERTOIRE
       BINCSOE  EQU   X'80'             EBCDIC
       BINCSOAI EQU   X'40'             ASCII / ISCII / ITA#5
       BINRSV30 EQU   X'20'             RESERVED
       BINRSV31 EQU   X'10'             RESERVED
       BINCSOC1 EQU   X'0C'             00= CODE 0 (MAIN CODE)
       *                                    SELECTION IS EBCDIC
       *                                01= CODE 0 (MAIN CODE)
       *                                    SELECTION IS ASCII /
       *                                    ISCII / ITA#5
       BINCSOC2 EQU   X'03'             00= CODE 1 (ALTERNATE
       *                                    CODE SELECTION IS
       *                                    EBCDIC
       *                                01= CODE 1 (ALTERNATE
       *                                    CODE SELECTION IS
       *                                    ASCII / ISCII /
       *                                    ITA#5
000017 BINGENCO DS    X                 GENERAL CHARACTERISTICS
       BINRSV32 EQU   X'C0'             RESERVED
       BINWSDF  EQU   X'20'             0 = PLU MAY SEND DATA
       *                                    FIRST
       *                                1 = SLU MUST SEND DATA
       *                                    FIRST
       BINRSV33 EQU   X'10'             RESERVED
       BINIAO   EQU   X'08'             0 = SLU WILL INITIATE
       *                                    ATTENDED
       *                                1 = SLU WILL INITIATE
       *                                    UNATTENDED
       BINAAO   EQU   X'04'             0 = SLU WILL NOT
       *                                    ALTERNATE BETWEEN
       *                                    ATTENDED AND
       *                                    UNATTENDED
       *                                1 = SLU MAY ALTERNATE
       *                                    BETWEEN ATTENDED AND
```

```
       *                                    UNATTENDED
       BINRSV34 EQU   X'03'               RESERVED
000018 BINRSV35 DS    X                   RESERVED
       *
       ********************************************************************
       *  OVERLAY FOR 'BINPSCHR' (PRESENTATION SERVICES CHARACTERISTICS
       *                          FOR PS PROFILE 6)
       ********************************************************************
000019         ORG   BINPSCHR
00000E BINLULEV DS    X                   LU-6 LEVEL
       BINLV02  EQU   X'02'               LEVEL 2
00000F BINRSV36 DS    XL6                 RESERVED
000015 BINFLG0  DS    X                   FLAGS
       BINDSSSP EQU   X'80'               DISTRIBUTED SYSTEMS SECURITY
       *                                    SUPPORTED
       *                                  0=EXTENDED SECURITY MECHANISMS
       *                                    ARE NOT SUPPORTED
       *                                  1=AT LEAST ONE SECURITY
       *                                    MECHANISM IS SUPPORTED
       BINDESS  EQU   X'40'               Extended Security Sense Codes
       *                                  0= Extended security sense
       *                                     codes will not be accepted on
       *                                     incoming FMH-7s
       *                                  1= Extended security sense
       *                                     codes will be accepted on
       *                                     incoming FMH-7s
000016 BINFLG1  DS    X                   LU 6.2 FLAGS
       BINCLSS  EQU   X'10'               ACCESS SECURITY SUBFIELD SUPPORT:
       *                                  0= ACCESS SECURITY INFORMATION
       *                                     FIELD WILL NOT BE ACCEPTED ON
       *                                     INCOMING FMH-5S
       *                                  1= ACCESS SECURITY INFORMATION
       *                                     FIELD WILL BE ACCEPTED ON
       *                                     INCOMING FMH-5S
       *
       BINSLAPS EQU   X'08'               SESSION LEVEL SECURITY PROTOCOL
       *
       *
       BINDPWS  EQU   X'04'               Password Substitution Support:
       *                                  0= Substituted passwords will
       *                                     not be accepted on incoming
       *                                     FMH-5s
       *                                  1= Substituted passwords will
       *                                     be accepted on incoming
       *                                     FMH-5s
       *
       BINAVFS  EQU   X'02'               ALREADY - VERIFIED FUNCTION
       *                                  SUPPORT
       *                                  0= ALREADY - VERIFIED FUNCTION
       *                                     WILL NOT BE ACCEPTED ON
       *                                     INCOMING FMH_5
       *                                  1= ALREADY - VERIFIED FUNCTION
       *                                     WILL BE ACCEPTED ON INCOMING
       *                                     FMH_5
       BINPV    EQU   X'01'               PERSISTENT VERIFICATION FUNCTION
       *                                  SUPPORT
       *                                  0= PERSISTENT VERIFICATION
       *                                     FUNCTION WILL NOT BE ACCEPTED
       *                                     ON INCOMING FMH_5
       *                                  1= PERSISTENT VERIFICATION
       *                                     FUNCTION WILL BE ACCEPTED ON
       *                                     INCOMING FMH_5
000017 BINFLG2  DS    X                   MORE LU 6.2 FLAGS
       BINSYNCH EQU   X'60'               SYNCHRONIZATION LEVEL:
       *    VALUES FOR BINSYNCH
       BINCONF  EQU   X'20'               CONFIRM SUPPORTED
       BINCSBK  EQU   X'40'               CONFIRM, SYNC POINT, AND
```

```
       *                                BACKOUT SUPPORTED
       BINRS    EQU   X'10'             RECONNECT SUPPORT:
       *                                0= RECONNECT NOT SUPPORTED
       *                                1= RECONNECT SUPPORTED
       BINRSR   EQU   X'0C'             RESPONSIBILITY FOR SESSION
       *                                REINITIATION:
       * NOTE: BINRSR IS RESERVED WHEN PARALLEL SESSIONS ARE SUPPORTED
       *       (I.E. WHEN BINPSS IS SET)
       *    VALUES FOR BINRSR
       BINOPRC  EQU   X'00'             OPERATOR CONTROLLED
       BINPRIMH EQU   X'04'             PRIMARY WILL REINITIATE
       BINSECNH EQU   X'08'             SECONDARY WILL REINITIATE
       BINETHR  EQU   X'0C'             EITHER MAY REINITIATE
       *
       BINPSS   EQU   X'02'             PARALLEL SESSION SUPPORT FOR
       *                                LU-LU PAIR:
       *                                0= PSS NOT SUPPORTED
       *                                1= PSS SUPPORTED
       BINGDSVF EQU   X'01'             CHANGE NUMBER OF SESSIONS GDS
       *                                VARIABLE FLOW SUPPORT:
       *                                0= NOT SUPPORTED
       *                                1= SUPPORTED
000018 BINFLG3  DS    X
       BINRSV37 EQU   X'80'             RESERVED
       BINLTDRC EQU   X'40'             1= LIMITED RESOURCE EXISTS
       BINRSV38 EQU   X'3C'             RESERVED
       BIN6CMP1 EQU   X'02'             APPLIES ONLY to BINSMED1
       *                                (SEE BINCMP1 AND BINCMP2)
       BIN6CMP2 EQU   X'01'             (SEE BINCMP1 AND BINCMP2)
       ***********************************************************************
       *  OVERLAY FOR 'BINPSCHR' (GENERIC OVERLAY FOR COMPRESSION)
       ***********************************************************************
000019         ORG   BINPSCHR
00000E BINRSV53 DS    XL10              UNCHANGED BY COMPRESSION
       *                                REFER TO SPECIFIC PROFILE
000018 BINCOMP  DS    XL1               COMPRESSION FLAG BYTE
       BINCMP1  EQU   X'02'             COMPRESSION BIT 1
       BINCMP2  EQU   X'01'             COMPRESSION BIT 2
       *                                THE COMBINATIONS:
       *                                    BIND REQUEST:
       *                                    00 - NO COMPRESSION
       *                                    01 - COMPRESSION BID
       *                                    10 - UNUSED
       *                                    11 - COMPRESSION REQUIRED
       *                                    BIND RESPONSE:
       *                                    10 - COMPRESSION USED
       *                                    XX - REMAINING - UNUSED
000019         ORG   ,                  END OF OVERLAYS
```

FMH-5 (ISTFM5)

```
  LOC     SOURCE STATEMENT
000000 ISTFM5   DSECT                    FMH5 MAPPING
000000 FM5BASE  DS    0CL10              FIXED LENGTH BASE
000000 FM5LENTH DS    X                  LENGTH FIELD
000001 FM5FLAG1 DS    X                  FLAG FIELDS 1
       FM5CONCT EQU   X'80'                CONCATENATION INDICATOR
       FM5TYP   EQU   X'7F'                FMH TYPE MASK
       FM5TYPE5 EQU   X'05'               IBM ARCHITECTED FMH5
000002 FM5TYPE  DS    XL2                FMH5 TYPE
       FM5ATTCH EQU   X'02FF'              FMH5 TYPE = ATTACH
000004 FM5FLAG2 DS    X                  FLAG BYTE
       FM5UIDAV EQU   X'80'                USER ID ALREADY VERIFIED
       FM5PV1   EQU   X'40'                USER ID SIGNED ON
       FM5PV2   EQU   X'20'                USER ID SIGN ON
       FM5PWS   EQU   X'10'                PASSWORD SUBSTITUTION
       *                                   IF THIS BIT IS 0 AND A PASSWORD
```

```
       *                                  IS PRESENT IT IS IN THE CLEAR
       *                                  IF THIS BIT IS 1 AND A PASSWORD
       *                                  IS PRESENT IT IS A SUBSTITUTED
       *                                  PASSWORD
       FM5PIPPR EQU   X'08'               PIP PRESENT AFTER FMH5
       FM5DSSPR EQU   X'04'               DISTRIBUTED SYSTEMS SECURITY
       *                                  AUTHENTICATION TOKEN GDS
       *                                  PRESENT AFTER FMH-5 (AND PIP
       *                                  GDS IF PRESENT).  IF THIS BIT
       *                                  IS ON, FM5UIDAV, FM5PV1, AND
       *                                  FM5PV2 MUST BE ZERO AS WELL AS
       *                                  THE SECURITY ACCESS SUBFIELDS.
       *
000005 FM5LNFLP DS    X                  LENGTH OF FIXED LENGTH PARAMETERS
000006 FM5FXLEN DS    0XL3               FIXED LENGTH PARAMETERS
000006 FM5RSCTP DS    X                  RESOURCE TYPE
       FM5BASIC EQU   X'D0'              BASIC CONVERSATION
       FM5MAPED EQU   X'D1'              MAPPED CONVERSATION
       FM5FDBAS EQU   X'D2'              FULL-DUPLEX BASIC CONVERSATION
       FM5FDMAP EQU   X'D3'              FULL-DUPLEX MAPPED
       *                                   CONVERSATION
000007          DS    C                  RESERVED
000008 FM5FLAG3 DS    X                  FLAGS FOR FIXED LENGTH PARMS
       FM5SYNCH EQU   X'C0'              SYNCHRONIZATION LEVEL MASK
       FM5NONE  EQU   X'00'                 NONE
       FM5CONFM EQU   X'40'                 CONFIRM
       FM5CSB   EQU   X'80'                 CONFIRM, SYNC POINT, BACKOUT
       FM5RESUP EQU   X'20'              RECONNECTION SUPPORT
000009 FM5LNTPN DS    X                  LENGTH OF TRANSACTION PROGRAM NAME
       *                                   (NOT INCLUDING THIS BYTE)
00000A FM5TPNAM DS    0X                 TRANSACTION PROGRAM NAME
       *
000000 FM5ASI   DSECT                    ACCESS SECURITY INFORMATION
       *                                    SUBFIELDS
000000 FM5LNASI DS    X                  LENGTH OF ASI SUBFIELDS
       *                                   (NOT INCLUDING THIS BYTE)
000001 FM5ASEC  DS    0X                 CONTAINS ALL ACCESS SECURITY
       *                                  SUBFIELDS.  THESE SUBFIELDS ARE
       *                                  MAPPED BY THE FM5ACCSE DSECT.
       *                                  THERE MAY BE ZERO OR MORE OF
       *                                  THESE SUBFIELDS, AND EACH MUST
       *                                  BE SEPARATELY MAPPED BY THE
       *                                  FM5ACCSE DSECT.
       *
000000 FM5LUOW1 DSECT                    LOGICAL UNIT OF WORK IDENTIFIER
       *                                   FIELD
000000 FM5LNLUW DS    X                  LENGTH OF LUOW ID
       *                                   (NOT INCLUDING THIS BYTE)
000001 FM5LUWI  DS    0X                 LUOW ID
000001 FM5LNFQN DS    X                  LENGTH OF FULLY QUALIFIED LU NAME
       *                                   (NOT INCLUDING THIS BYTE)
000002 FM5FQNAM DS    0X                 FULLY QUALIFIED LU NAME
       *
000000 FM5LUOW2 DSECT                    LUOW
000000 FM5LUWIN DS    XL6                LUOW INSTANCE NUMBER
000006 FM5LUWSN DS    XL2                LUOW SEQUENCE NUMBER
       *
000000 FM5CVCOR DSECT                    CONVERSATION CORRELATOR
000000 FM5LNCCS DS    X                  LENGTH OF CONVERSATION CORRELATOR
       *                                   OF SENDER
       *                                   (NOT INCLUDING THIS BYTE)
000001 FM5CCS   DS    0X                 CONVERSATION CORRELATOR OF SENDING
       *                                   TRANSACTION
       *
000000 FM5SEQNM DSECT                    SEQUENCE NUMBER MAP
000000 FM5LNSNM DS    X                  LENGTH OF SEQUENCE NUMBER
       *                                   (NOT INCLUDING THIS BYTE)
```

```
000001 FM5SNM   DS    XL8                  SEQUENCE NUMBER
       *
       ***********************************************************************
       * ACCESS SECURITY SUBFIELD                                            *
       *                                                                     *
       * THIS DSECT IS USED TO MAP EACH ACCESS SECURITY SUBFIELD.  THESE     *
       * SUBFIELDS ARE ALL CONTAINED IN THE FIELD 'FM5ASEC'.  YOU MUST       *
       * DETERMINE HOW MANY SUBFIELDS ARE SPECIFIED, AND DETERMINE THE       *
       * LENGTH OF EACH OF THE SUBFIELDS.                                    *
       *                                                                     *
       ***********************************************************************
000000 FM5ACCSE DSECT                      ACCESS SECURITY SUBFIELD
000000 FM5ASLL  DS    X                    SUBFIELD LENGTH
       *                                     (NOT INCLUDING THIS BYTE)
000001 FM5ASTY  DS    X                    SUBFIELD TYPE
       FM5ASIPR EQU   X'00'                  PROFILE
       FM5ASIPW EQU   X'01'                  PASSWORD
       FM5ASIID EQU   X'02'                  USER ID
000002 FM5ASDA  DS    0X                   SUBFIELD DATA
       *
       ***********************************************************************
       *   PROGRAM INITIALIZATION PARAMETER (PIP).                           *
       *                                                                     *
       *   THE PIP, IF IT EXISTS (INDICATED BY FM5PIPPR), FOLLOWS            *
       *   THE FMH5.                                                         *
       *                                                                     *
       *   ADDRESSABILITY: IF PIP EXISTS, PIP LOCATED AFTER FMH5.            *
       *                                                                     *
       ***********************************************************************
000000 FM5PIPFM DSECT                      PIP FORMAT
000000 FM5PIPLN DS    XL2                  PIP LENGTH (INCLUDING THIS BYTE)
000002 FM5PIPGD DS    XL2                  GDS INDICATOR
       FM5PIPF5 EQU   X'12F5'                PIP VARIABLE
000004 FM5PIPSF DS    0X                   ZERO OR MORE PIP SUBFIELDS, EACH OF
       *                                     WHICH HAS THE FOLLOWING FORMAT
000000 FM5PIPSM DSECT                      PIP SUBFIELD MAP
000000 FM5PIPSL DS    XL2                  SUBFIELD LENGTH
       *                                     (INCLUDING THIS BYTE)
000002 FM5PIPSG DS    XL2                  GDS INDICATOR
       FM5PIPE2 EQU   X'12E2'                PIP SUBFIELD
000004 FM5PIPSD DS    0X                   SUBFIELD DATA
```

RPL Extension (ISTRPL6X)

```
  LOC     SOURCE STATEMENT
000000 ISTRPL6X DSECT
000000 RPL6AREA DS    0CL112             START OF APPC EXTENSION
000000 RPL6CBID DS    CL4                CONTROL BLOCK IDENTIFIER
000004 RPL6REQ  DS    XL1                TYPE OF APPCCMD
000005 RPL6QUAL DS    XL1                SUBTYPE OF APPCCMD
000006          DS    XL2                RESERVED
000008 RPL6CNVD DS    XL4                CONVERSATION ID
00000C RPL6USR  DS    XL4                USER FIELD
000010 RPL6SNSO DS    XL4                SENSE DATA SPECIFIED ON APPCCMD
000014 RPL6SNSI DS    XL4                SENSE DATA RETURNED BY APPCCMD
000018 RPL6SGNL DS    XL4                SIGNAL DATA RETURNED
00001C RPL6SIDL DS    XL1                LENGTH OF SESSION ID
00001D          DS    XL3                RESERVED
000020 RPL6SSID DS    XL8                SESSION IDENTIFICATION
000028 RPL6RC   DS    0XL4               RPL6 RETURN CODE
000028 RPL6RCPR DS    XL2                PRIMARY RETURN CODE
00002A RPL6RCSC DS    XL2                SECONDARY RETURN CODE
00002C RPL6FLGS DS    0XL4               INDICATORS SPECIFIC TO VTAM'S
       *                                 APPCCMD MACRO
00002C RPL6FLG1 DS    XL1                FIRST INDICATORS BYTE
       RPL6FILL EQU   X'80'              FILL INDICATOR
       RPL6CD   EQU   X'40'              CD KEYWORD INDICATION
```

```
|                                *        EQU   X'20'               RESERVED
|                                RPL6SLS  EQU   X'10'               PARTNER LU VERIFICATION
|                                *                                  INDICATOR
|                                RPL6CFTX EQU   X'08'               CONFTXT INDICATOR
|                                RPL6LIST EQU   X'06'               SCOPE OF INFORMATION TO BE
|                                *                                  RETURNED IN THE RESTORE COMMAND
|                                *        EQU   X'01'               RESERVED
|                         00002D RPL6FLG2 DS    XL1                SECOND INDICATORS BYTE
|                                *        EQU   X'80'               RESERVED
|                                RPL6RTSX EQU   X'40'               RTS_RCVD RETURN  1=EXPD,0=BOTH
|                                RPL6CXMD EQU   X'30'               CONXMOD INDICATORS
|                                RPL6TYPE EQU   X'0C'               TYPE INDICATOR
|                                RPL6NAMU EQU   X'03'               NAME USE REQUESTED WHEN APPLICATION
|                                *                                  IS ACTING AS A GENERIC RESOURCE
|                         00002E RPL6FLG3 DS    XL1                THIRD INDICATORS BYTE
|                                RPL6LOCK EQU   X'80'               LOCKS INDICATOR
|                                RPL6DERC EQU   X'60'               DEACTIVATION REASON CODE
|                                RPL6EXDR EQU   X'10'               EXPEDITED DATA RECEIVED
|                                RPL6CMOD EQU   X'0C'               CONMODE INDICATOR
|                                RPL6LAST EQU   X'03'               LAST INDICATOR
|                         00002F RPL6FLG4 DS    XL1                FOURTH INDICATORS BYTE
|                                RPL6AFFN EQU   X'C0'               GENERIC RESOURCE AFFINITY OWNER
|                                *        EQU   X'3F'               RESERVED
|                         000030 RPL6LU   DS    CL8                NAME OF LU
|                         000038 RPL6MODE DS    CL8                MODE NAME
|                         000040 RPL6WHAT DS    0XL2
|                         000040 RPL6RCV1 DS    XL1                WHAT RECEIVED INDICATOR
|                                RPL6WDAT EQU   X'80'               WHATRCV=DATA
|                                RPL6WDAC EQU   X'40'               WHATRCV=DATA_COMPLETE
|                                RPL6WDAI EQU   X'20'               WHATRCV=DATA_INCOMPLETE
|                                RPL6WSND EQU   X'10'               WHATRCV=SEND
|                                RPL6WCFM EQU   X'08'               WHATRCV=CONFIRM
|                                RPL6WDAL EQU   X'04'               WHATRCV=DEALLOCATE
|                                RPL6WLOG EQU   X'02'               WHATRCV=LOG_DATA
|                                RPL6WPSH EQU   X'01'               WHATRCV=PS_HEADER
|                         000041 RPL6RCV2 DS    XL1                RESERVED FOR BIT MASK FOR THE
|                                RPL6WPSI EQU   X'80'               WHATRCV=PARTIAL_PS_HEADER
|                                *        EQU   X'7F'               NOT USED
|                         000042 RPL6RTUN DS    XL1                RETURNED INDICATORS AS
|                                *                                  A RESULT OF APPCCMD
|                                RPL6RMH5 EQU   X'80'               FMH5RCV INDICATOR
|                                RPL6RLOG EQU   X'40'               LOGRCV INDICATOR
|                                RPL6RSIG EQU   X'20'               SIGRCV INDICATOR
|                                RPL6CLSA EQU   X'10'               PARTNER LU ACCEPTS SECURITY
|                                *                                  SUBFIELDS ON FMH5
|                                RPL6AVFA EQU   X'08'               PARTNER LU ACCEPTS REQUESTS FOR
|                                *                                  ALREADY VERIFIED
|                                RPL6PV   EQU   X'04'               PARTNER LU ACCEPTS REQUESTS FOR
|                                *                                  PERSISTENT VERIFICATION
|                                RPL6CRYP EQU   X'03'               ENCRYPTION LEVEL
|                         000043 RPL6MH5L DS    XL1                LENGTH OF THE FMH 5 RECEIVED
|                         000044 RPL6CCST DS    XL1                CURRENT CONVERSATION STATE
|                         000045 RPL6ACTV DS    XL1                RPL6 ACTIVE INDICATOR
|                                *                                  FF=ACTIVE / 00=INACTIVE
|                         000046 RPL6DETP DS    XL1                DEACTIVATION TYPE CODE
|                         000047 RPL6EXDL DS    XL1                LENGTH OF EXPEDITED DATA
|                         000048 RPL6TID  DS    0A                 TASK ID
|                         000048 RPL6MID  DS    XL2                MACHINE ID
|                         00004A RPL6TIX  DS    XL2                TASK INDEX OF CURRENTLY
|                                *                                  EXECUTING TASK
|                         00004C RPL6RPL  DS    A                  POINTER BACK TO THE RPL
|                         000050 RPL6STBF DS    A                  POINTER TO CURRENT BUFFER
|                                *                                  AT STORAGE SHORTAGE
|                         000054 RPL6STDS DS    A                  DISPLACEMENT IN CURRENT
|                                *                                  BUFFER AT STORAGE SHORTAGE
|                         000058 RPL6VAOL DS    XL2                RESERVED
|                         00005A RPL6VAIL DS    XL2                VTAM-TO-APPL VECTORLIST AREA LENGTH
```

```
                 *
00005C RPL6NET  DS    CL8                 NQN NETID
000064 RPL6CGID DS    XL4                 CONVERSATION GROUP ID
000068 RPL6VAOA DS    A                   RESERVED
00006C RPL6VAIA DS    A                   VTAM-to-APPL VECTORLIST AREA
       RPL6END  EQU   *                   END OF RPL6
       ************************************************************************
       *                                                                      *
       *   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6AFFN.        *
       *   THEY REPRESENT THE "LUAFFIN=" VALUE.                               *
       *                                                                      *
       ************************************************************************
       RPL6NAFF EQU   X'00'               LUAFFIN NOT SPECIFIED
       RPL6NAAF EQU   X'40'               LUAFFIN=NOTAPPL
       RPL6AAFF EQU   X'80'               LUAFFIN=APPL
       ************************************************************************
       *                                                                      *
       *   THE FOLLOWING CONSTANT VALUES ARE THOSE SPECIFIED IN THE           *
       *   EXPEDITED DATA FLOW CONTROL RU "SIGNAL".                           *
       *                                                                      *
       ************************************************************************
       RPL6SIG1 EQU   X'00010001'         SIGNAL DATA RETURNED TO APPLICATION
       ***********************************************************************
       *                                                                     *
       *   THE FOLLOWING CONSTANT IS DEFINED AS A SYMBOLIC REFERENCE         *
       *   TO THE APPC CONTROL BLOCK ID (RPL6).                              *
       *                                                                     *
       ***********************************************************************
       RPL6ID   EQU   C'APPC'             APPC CONTROL BLOCK ID
       *
       ************************************************************************
       *                                                                      *
       *   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6REQ.         *
       *   THEY REPRESENT THE "CONTROL=" VALUE.                               *
       *                                                                      *
       ************************************************************************
       RPL6ALLC EQU   X'10'               ALLOC
       RPL6PLOC EQU   X'11'               PREALLOC
       RPL6SFM5 EQU   X'12'               SENDFMH5
       RPL6RSRV EQU   X'20'               RESETRCV
       RPL6DEAL EQU   X'30'               DEALLOC
       RPL6DEAQ EQU   X'31'               DEALLOCQ
       RPL6OPER EQU   X'40'               OPRCNTL
       RPL6PREC EQU   X'50'               PREPRCV
       RPL6RFH5 EQU   X'60'               RCVFMH5
       RPL6RCV  EQU   X'70'               RECEIVE
       RPL6RCVX EQU   X'71'               RCVEXPD
       RPL6RJCT EQU   X'80'               REJECT
       RPL6SEND EQU   X'90'               SEND
       RPL6SNDX EQU   X'91'               SENDEXPD
       RPL6SETS EQU   X'A0'               SETSESS
       RPL6TSTS EQU   X'B0'               TESTSTAT
       ************************************************************************
       *                                                                      *
       *   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6QUAL.        *
       *   THEY REPRESENT THE "QUALIFY=" VALUE.                               *
       *                                                                      *
       ************************************************************************
       RPL6NQUA EQU   X'00'               NULL QUALIFER
       RPL6APRG EQU   X'01'               ABNDPROG
       RPL6ASRV EQU   X'02'               ABNDSERV
       RPL6ATIM EQU   X'03'               ABNDTIME
       RPL6AUSR EQU   X'04'               ABNDUSER
       RPL6ANY  EQU   X'05'               ANY
       RPL6CNOS EQU   X'06'               CNOS
       RPL6CFRM EQU   X'07'               CONFIRM
       RPL6CFMD EQU   X'08'               CONFRMD
```

```
RPL6DATA EQU   X'09'              DATA
RPL6DCON EQU   X'0A'              DATACON
RPL6DFLU EQU   X'0B'              DATAFLU
RPL6DFIN EQU   X'0C'              DEFINE
RPL6DSPY EQU   X'0D'              DISPLAY
RPL6ERR  EQU   X'0E'              ERROR
RPL6FLSH EQU   X'0F'              FLUSH
RPL6RQSD EQU   X'10'              RQSEND
RPL6SPEC EQU   X'11'              SPEC
RPL6ACT  EQU   X'12'              ACTSESS
RPL6DACT EQU   X'13'              DACTSESS
RPL6ALCD EQU   X'14'              ALLOCD
RPL6IMED EQU   X'15'              IMMED
RPL6CWIN EQU   X'16'              CONWIN
RPL6SESN EQU   X'17'              SESSION
RPL6CONV EQU   X'18'              CONV
RPL6SUSP EQU   X'19'              SUSPEND
RPL6RESM EQU   X'1A'              RESUME
RPL6REST EQU   X'1B'              RESTORE
RPL6SYNB EQU   X'1C'              SYNCBEG
RPL6SYNE EQU   X'1D'              SYNCEND
RPL6CNGP EQU   X'1E'              CONVGRP
RPL6SESF EQU   X'1F'              WHENFREE
RPL6IANY EQU   X'20'              IANY
RPL6ISPC EQU   X'21'              ISPEC
RPL6QALL EQU   X'22'              ALL
RPL6IALL EQU   X'23'              IALL
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6FILL.       *
*   THEY REPRESENT THE "FILL=" VALUE.                                 *
*                                                                     *
***********************************************************************
RPL6BUFF EQU   X'00'              BUFF
RPL6LL   EQU   X'80'              LL
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6CD          *
*   THEY REPRESENT THE "CD=" VALUE                                    *
*                                                                     *
***********************************************************************
RPL6CDIM EQU   X'00'              "CD=IMMED"
RPL6CDDE EQU   X'40'              "CD=DEFER"
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6CFTX.       *
*   THEY REPRESENT THE "CONFTXT=" VALUE.                              *
*                                                                     *
***********************************************************************
RPL6CFT  EQU   X'08'              YES
RPL6NCFT EQU   X'00'              NO
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6TYPE.       *
*   THEY REPRESENT THE "TYPE=" VALUE.                                 *
*                                                                     *
***********************************************************************
RPL6TBIT EQU   X'0C'              TYPE BITS POSITION
RPL6USER EQU   X'0C'              USER
RPL6PRGM EQU   X'04'              PROGRAM
RPL6SVC  EQU   X'08'              SERVICE
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6NAMU.       *
*   THEY REPRESENT THE "NAMEUSE=" VALUE.                              *
*                                                                     *
***********************************************************************
```

```
RPL6NUNS EQU   X'00'               NAME USAGE NOT SPECIFIED
RPL6ANAM EQU   X'01'               WHEN APPLICATION IS ACTING AS A
*                                  GENERIC RESOURCE, SESSIONS STARTED
*                                  WITH PARTNER LU SHOULD USE THE
*                                  APPLICATION NETWORK NAME
RPL6GNAM EQU   X'02'               WHEN APPLICATION IS ACTING AS A
*                                  GENERIC RESOURCE, SESSIONS STARTED
*                                  WITH PARTNER LU SHOULD USE GENERIC
*                                  NAME OF THE APPLICATION
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6LOCK.       *
*   THEY REPRESENT THE "LOCKS=" VALUE.                                *
*                                                                     *
***********************************************************************
RPL6LONG EQU   X'00'               LONG
RPL6SHRT EQU   X'80'               SHORT
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6DERC.       *
*                                                                     *
***********************************************************************
RPL6RNRM EQU   X'00'               NORMAL
RPL6RABN EQU   X'40'               ABNORMAL
RPL6RANR EQU   X'60'               ABNORMAL, NORETRY
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6CMOD.       *
*   THEY REPRESENT THE "CONMODE=" VALUE.                              *
*                                                                     *
***********************************************************************
RPL6CBIT EQU   X'0C'               CONMODE BITS POSITION
RPL6LLCA EQU   X'00'               LLCA
RPL6BFCA EQU   X'04'               BUFFCA
RPL6CS   EQU   X'08'               CS
RPL6SAME EQU   X'0C'               SAME
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6CXMD.       *
*   THEY REPRESENT THE "CONXMOD=" VALUE.                              *
*                                                                     *
***********************************************************************
RPL6CXBT EQU   X'30'               CONXMOD BIT POSITIONS
RPL6CSCX EQU   X'00'               CS
RPL6CACX EQU   X'10'               CA
RPL6SAMX EQU   X'30'               SAME
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6LAST.       *
*                                                                     *
***********************************************************************
RPL6NLST EQU   X'00'               SESSIONS EXIST FOR THE
*                                  SPECIFIED MODE
RPL6LMOD EQU   X'01'               LAST SESSION DEACTIVATED
*                                  FOR THE SPECIFIED MODE
RPL6NCTL EQU   X'02'               LAST SESSION DEACTIVATED
*                                  FOR NON-CONTROL MODES
RPL6ALL  EQU   X'03'               ALL SESSIONS FOR THIS LU
*                                  HAVE BEEN DEACTIVATED
***********************************************************************
*                                                                     *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6LIST.       *
*                                                                     *
***********************************************************************
RPL6LINO EQU   X'00'               NO INFORMATION RETURNED
RPL6LINS EQU   X'02'               LU NAME, MODE NAME, AND LM
*                                  TABLE INFORMATION RETURNED
```

```
RPL6LIAL EQU   X'04'               ALL INFORMATION IN RESTORE
*                                  STRUCTURE RETURNED
**********************************************************************
*                                                                    *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6CCSL.      *
*   THEY REPRESENT THE CURRENT CONVERSATION STATE.                   *
*                                                                    *
**********************************************************************
RPL6RSET EQU   X'00'               RESET  (INITIAL STATE)
*   ** HALF-DUPLEX CONVERSATION STATES **
RPL6SND  EQU   X'01'               SEND
RPL6RECV EQU   X'02'               RECEIVE
RPL6RVCF EQU   X'03'               RECEIVE CONFIRM
RPL6RVCS EQU   X'04'               RECEIVE CONFIRM SEND
RPL6RVCD EQU   X'05'               RECEIVE CONFIRM DEALLOCATE
RPL6PNDD EQU   X'06'               PEND DEALLOCATE
RPL6PECL EQU   X'07'               PEND END CONVESATION LOG
RPL6ENDC EQU   X'08'               END CONVERSATION  (FINAL)
RPL6PNDS EQU   X'09'               PENDING SEND
RPL6PRVL EQU   X'0A'               PENDING RCV LOG
*   ** FULL-DUPLEX CONVERSATION STATES **
RPL6FDRS EQU   X'80'               FDX RESET  (FINAL)
RPL6FDSR EQU   X'81'               FDX SEND/RECEIVE
RPL6FDSO EQU   X'82'               FDX SEND-ONLY
RPL6FDRO EQU   X'83'               FDX RECEIVE-ONLY
RPL6FSRL EQU   X'84'               FDX PENDING SEND/RCV LOG
RPL6FROL EQU   X'85'               FDX PENDING RCV-ONLY LOG
RPL6FRSL EQU   X'86'               FDX PENDING RESET LOG
*    ** PENDING CONVERSATION ALLOCATION **
RPL6PALC EQU   X'FF'               PENDING ALLOCATE
**********************************************************************
*                                                                    *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6DETP.      *
*   THEY REPRESENT THE "DEACTYP=" VALUE.                             *
*                                                                    *
**********************************************************************
RPL6TCLP EQU   X'0F'               CLEANUP
RPL6TPVL EQU   X'FE'               PROTOCOL VIOLATION
**********************************************************************
*                                                                    *
*   THE FOLLOWING CONSTANT VALUES WILL BE RECORDED IN RPL6CRYP.      *
*   THEY REPRESENT THE ENCRYPTION LEVEL.                             *
*                                                                    *
**********************************************************************
RPL6CNON EQU   X'00'               NONE
RPL6CSEL EQU   X'01'               SELECTIVE DATA ENCRYPTION
RPL6CMAN EQU   X'03'               MANDATORY DATA ENCRYPTION
```

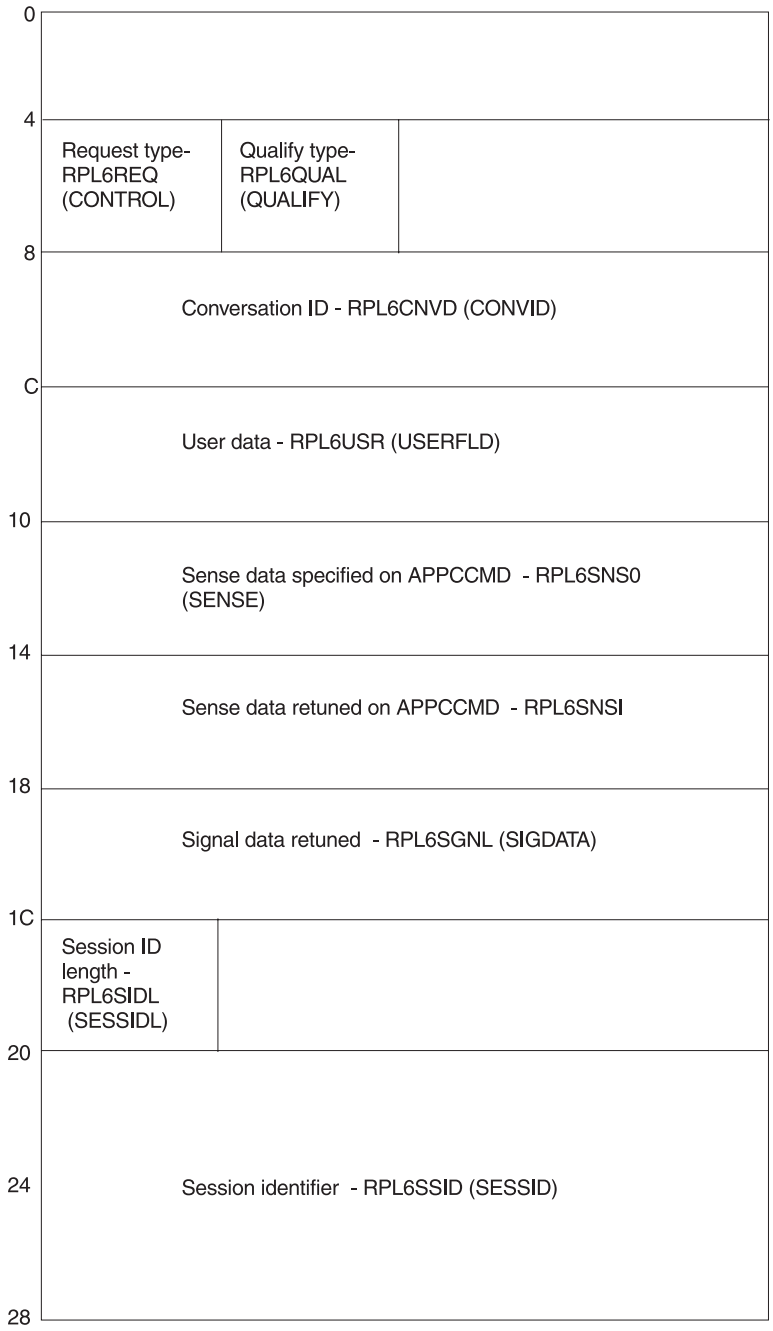

Figure 2. Layout of the RPL Extension (Part 1 of 3)

28	Primary return code - RPL6RCPR (RCPRI)		Secondary return code - RPL6RCSC (RCSEC)	
2C	Flag byte- RPL6FLG1 (FILL, CD SLS, CONFTXT, LIST)	Flag byte- RPL6FLG2 (RTSRTRN CONXMOD,TYPE, NAMEUSE)	Flag byte- RPL6FLG3 (LOCKS, DERC EXDR, CONMODE LAST)	Flag byte- RPL6FLG4 (LUAFFIN)
30	LU name - RPL6LU (LUNAME)			
34				
38	Mode name - RPL6MODE (LOGMODE)			
3C				
40	What-received field - RPL6RCV1 (WHATRCV)	What-received field - RPL6RCV2 (WHATRCV)	Returned bits- RPL6RTUN (FMH5RCV, LOGCRV, SIGRCV, CONVSECP, AFVA PRSISTVP, CRYPTLVL)	Received (FMH5 length - RPL6MH5L (FMH5LEN)
44	Current conversation state RPL6CCST (CONSTATE)	RPL in use - RPL6ACTV	Session deactivation - type code - RPL6DEPT (DEACTYP)	Expedited type code - RPL6EXDL
48	Task ID (The sublevel names are referenced by the VM system) - RPL6TID			
4C				

Figure 2. Layout of the RPL Extension (Part 2 of 3)

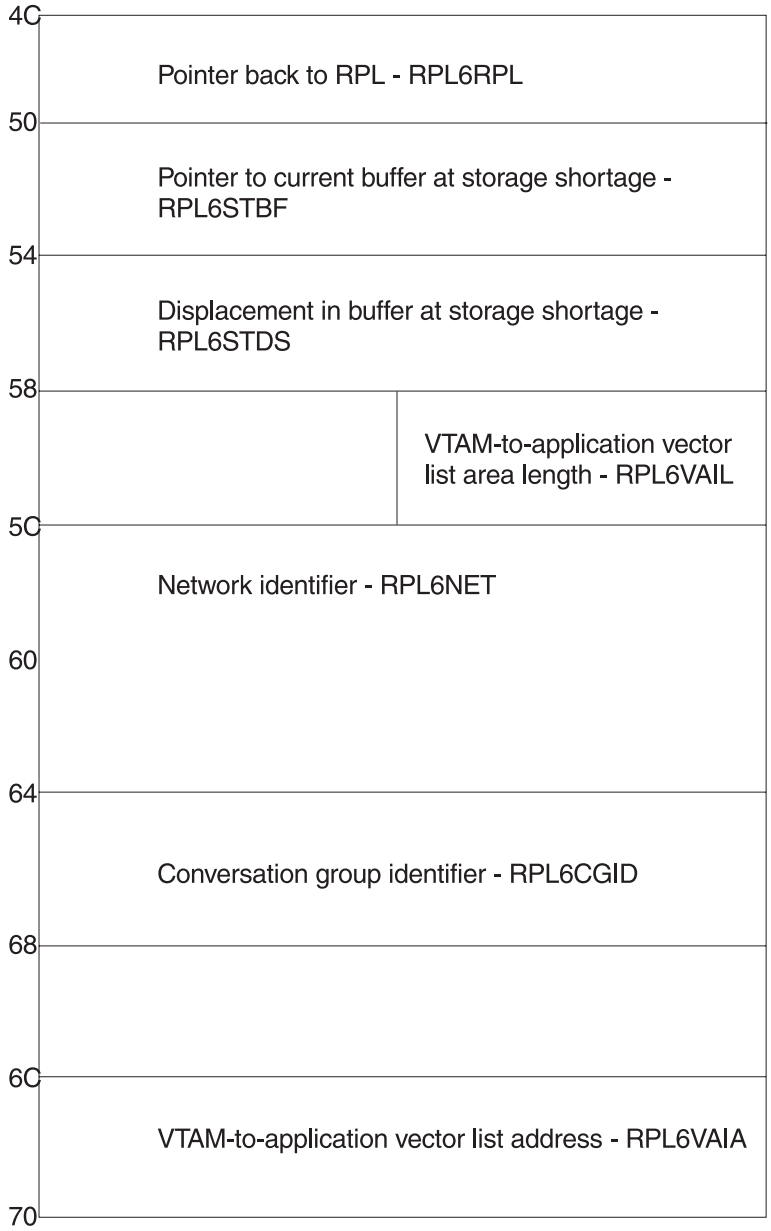

Figure 2. Layout of the RPL Extension (Part 3 of 3)

CNOS Session Limits Data Structure (ISTSLCNS)

```
  LOC     SOURCE STATEMENT
000000 ISTSLCNS DSECT                      SESSION LIMITS FOR CNOS
000000 SLCSESSL DS    H                    SESSION LIMIT
000002 SLCMCWL  DS    H                    MINIMUM NUMBER OF CONTENTION WINNER
       *                                   SESSIONS - LOCAL LU
000004 SLCMCWP  DS    H                    MIMIMUM NUMBER OF CONTENTION WINNER
       *                                   SESSIONS - PARTNER LU
000006 SLCPARMS DS    XL1                  CNOS PARAMETERS
       SLCDRAL  EQU   X'80'                DRAINING OF LOCAL LU REQUESTED
       SLCDRAP  EQU   X'40'                DRAINING OF PARTNER LU REQUESTED
       SLCPRSPL EQU   X'20'                RESPONSIBLE FOR DEACTIVATION
       *                                   IF SET, PARTNER LU IS RESPONSIBLE
       *                                   IF NOT SET, LOCAL LU IS RESPONSIBLE
       SLCALL   EQU   X'10'                INDICATES IF CNOS IS FOR ONE MODE
       *                                   OR ALL MODES
```

```
        *                                   IF SET, CNOS IS FOR ALL MODES
        *                                   IF NOT SET, CNOS IS FOR ONE MODE
        SLCSSLU  EQU   X'08'                INDICATE IF THE PARTNER LU IS ONLY
        *                                   SINGLE SESSION CAPABLE
        *
        SLCLCLSI EQU   X'07'                LOCAL LU SECURITY SUBFIELD
        *                                   ACCEPTANCE INFORMATION
        SLCLNONE EQU   X'00'                NONE ACCEPTED
        SLCLCONV EQU   X'04'                SECURITY SUBFIELDS ACCEPTED
        *                                   ON FMH5
        SLCLAVFA EQU   X'02'                ALREADY VERIFIED REQUESTS ACCEPTED
        *
        SLCLPV   EQU   X'01'                PERSISTENT VERIFICATION REQUESTS
        *                                   ACCEPTED
000007          DS    XL1                   RESERVED
000008 SLCDSESL DS    H                     DEFINED SESSION LIMIT
00000A SLCDMCWL DS    H                     DEFINED MINIMUM NUMBER OF
        *                                   CONTENTION WINNER SESSIONS
        *                                   - LOCAL LU
00000C SLCDMCWP DS    H                     DEFINED MINUMUM NUMBER OF
        *                                   CONTENTION WINNER SESSIONS
        *                                   - PARTNER LU
00000E SLCPARM2 DS    XL1                   INDICATORS
        SLCDDRAL EQU   X'80'                DEFINED DRAINING OF LOCAL LU
        SLCDRSPL EQU   X'40'                DEFINED ACCEPTANCE DEACTIVATION
        *                                   RESPONSIBILITY OF LOCAL LU
        SLCDEFND EQU   X'20'                ON-ATTN CNOS DRIVEN DUE TO MODIFY
        *                                   DEFINE. OFF- ATTN CNOS DRIVEN DUE
        *                                   TO CNOS PROCESSED ON TARGET SIDE.
        *        EQU   X'10'-X'01'          RESERVED
00000F          DS    XL1                   RESERVED
        *
000010 SLCEND   DS    0X                    END OF ISTSLCNS
```

DEFINE/DISPLAY Session Limits Data Structure (ISTSLD)

```
  LOC     SOURCE STATEMENT
000000 ISTSLD   DSECT                       SESSION LIMITS - DEFINE/DISPLAY
        *
        * BEGINING OF LU SPECIFIC FIELDS
        *
000000 SLDLUPAR DS    0XL40                 LU SPECIFIC FIELDS
000000 SLDLU1   DS    XL1                   LU SPECIFIC FIELDS - BYTE 1
        SLDSCAP  EQU   X'C0'                LU'S SESSION CAPABILITY MASK
        SLDPARR  EQU   X'C0'                PARALLEL SESSION CAPABLE
        SLDPNDGP EQU   X'80'                PENDING PARALLEL STATE
        SLDPNDGS EQU   X'40'                PENDING SINGLE STATE
        SLDSINGL EQU   X'00'                SINGLE SESSION CAPABLE
        SLDSYNCH EQU   X'30'                NEGOTIATED LEVEL OF SYNCHRONIZATION
        *
        SLDCSBK  EQU   X'20'                CONFIRM, SYNC POINT, BACKOUT
        *                                   SUPPORTED
        SLDCONF  EQU   X'10'                CONFIRM SUPPORTED
        SLDSYNRT EQU   X'00'                SYNCHRONIZATION LEVEL NOT SET
        *        EQU   X'0F'                RESERVED
000001 SLDLU2   DS    XL1                   LU SPECIFIC FIELDS - BYTE 2
        SLDCLSV  EQU   X'80'                SECURITY SUBFIELD ACCEPTANCE
        *                                   INFORMATION IS VALID
        SLDPCLSA EQU   X'40'                PARTNER LU ACCEPTS SECURITY
        *                                   SUBFIELDS ON FMH5
        SLDPAVFA EQU   X'20'                PARTNER LU ACCEPTS REQUESTS FOR
        *                                   ALREADY VERIFIED FUNCTION
        SLDPPV   EQU   X'10'                PARTNER LU ACCEPTS REQUESTS FOR
        *                                   PERSISTENT VERIFICATION
        SLDLCLSA EQU   X'08'                LOCAL LU ACCEPTS SECURITY
        *                                   SUBFIELDS ON FMH5 FROM THIS
        *                                   PARTNER LU
```

```
       SLDLAVFA EQU   X'04'                LOCAL LU ACCEPTS REQUESTS FOR
       *                                   ALREADY VERIFIED FUNCTION
       SLDLPV   EQU   X'02'                LOCAL LU ACCEPTS REQUESTS FOR
       *                                   PERSISTENT VERIFICATION
       *        EQU   X'01'                RESERVED
000002 SLDFQNLN DS    XL1                  LENGTH OF FULLY QUALIFIED
       *                                   PARTNER LU NAME
000003 SLDFQNAM DS    XL17                 FULLY QUALIFIED PARTNER LU
       *                                   NAME - USE SLDFQNLN FOR
       *                                   ACTUAL LENGTH OF NAME
000014 SLDLU3   DS    XL1                  LU-SPECIFIC INDICATORS
       SLDCNVCP EQU   X'C0'                CONVERSATION CAPABILITY FIELD
       SLDCNVFD EQU   X'80'                FULL-DUPLEX OR HALF-DUPLEX,
       *                                   EXPEDITED DATA ALLOWED
       SLDCNVHD EQU   X'40'                HALF-DUPLEX CONVERSATIONS ONLY
       SLDCNVUN EQU   X'00'                CAPABILITY IS UNKNOWN
       *        EQU   X'20' - X'01'        RESERVED
000015 SLDNMUSE DS    XL1                  APPLICATION NAME USE FIELD,
       *                                   INDICATES FORM OF LU NAME USED BY
       *                                   LOCAL LU FOR SESSIONS WITH THIS
       *                                   PARTNER LU
       SLDNMUUN EQU   X'00'                NAME USE NOT YET KNOWN
       SLDNMUUV EQU   X'01'                USERVAR NAME IS BEING USED
       SLDNMUAN EQU   X'02'                APPL NETWORK NAME BEING USED
       SLDNMUGN EQU   X'03'                GENERIC NAME IS BEING USED
000016 SLDTYPE  DS    XL1                  TYPE OF LU ENTRY
       SLDSUPNM EQU   X'00'                SUPPLIED_NAME ENTRY
       SLDRCVNM EQU   X'01'                RCVD_NAME ENTRY
       SLDVARNM EQU   X'02'                VARIANT_NAME ENTRY
       SLDUNUNM EQU   X'03'                UNUSABLE_NAME ENTRY
       SLDDISNM EQU   X'04'                DISASSOC_NAME ENTRY
000017          DS    XL17                 RESERVED
       *
       *
       * END OF LU SPECIFIC FIELDS
       *
       * BEGINING OF MODE SPECIFIC FIELDS
       *
000028 SLDDSESL DS    H                    DEFINED SESSION LIMIT
00002A SLDDMCWL DS    H                    DEFINED NUMBER OF CONTENTION WINNER
       *                                   SESSIONS -- LOCAL LU
00002C SLDDMCWP DS    H                    DEFINED NUMBER OF CONTENTION WINNER
       *                                   SESSIONS -- PARTNER LU
00002E SLDDEFPA DS    XL1                  DEFINED PARAMETERS
       SLDDRSPL EQU   X'80'                DEFINED ACCEPTANCE OF DEACTIVATION
       *                                   RESPONSIBILITY, IF SET THEN THE
       *                                   LOCAL LU WILL ACCEPT RESPONSIBILITY
       SLDDDRAL EQU   X'40'                DEFINED ACCEPTANCE OF REQUEST TO
       *                                   DRAIN QUEUED ALLOCS, IF SET THEN
       *                                   LOCAL LU WILL ACCEPT THE REQUEST
       SLDDELET EQU   X'20'                MODE DELETION INDICATOR, IF SET
       *                                   APPL WILL ALLOW DELETION OF MODE
       SLDAUTOS EQU   X'10'                AUTOSES SPECIFIED AS ON DEFINE
       SLDMDSUS EQU   X'08'                MODE PENDING RECOVERY
       *        EQU   X'04'-X'01'          RESERVED
00002F SLDCNSPA DS    XL1                  CNOS PARAMETERS
       SLDDRAL  EQU   X'80'                DRAINING OF LOCAL LU
       SLDDRAP  EQU   X'40'                DRAINING OF PARTNER LU
       *        EQU   X'20'-X'01'          RESERVED
000030 SLDSESSL DS    H                    SESSION LIMIT
000032 SLDMCWL  DS    H                    MINIMUM NUMBER OF CONTENTION WINNER
       *                                   SESSIONS -- LOCAL LU
000034 SLDMCWP  DS    H                    MINIMUM NUMBER OF CONTENTION WINNER
       *                                   SESSIONS -- PARTNER LU
000036 SLDAUTO  DS    H                    AUTO ACTIVATE LIMIT
000038 SLDSESSC DS    H                    CURRENT SESSION COUNT
00003A SLDWINLC DS    H                    NUMBER OF CURRENT CONTENTION WINNER
```

```
       *                                    SESSIONS -- LOCAL LU
00003C SLDWINPC DS    H                     NUMBER OF CURRENT CONTENTION WINNER
       *                                    SESSIONS -- PARTNER LU
00003E SLDFREEC DS    H                     NUMBER OF FREE SESSIONS
000040 SLDQALLC DS    H                     NUMBER OF ALLOCATE REQUEST WAITING
       *                                    FOR FREE SESSIONS
000042          DS    XL2                   RESERVED
       *
       * END OF MODE SPECIFIC FIELDS
       *
000044 SLDEND   DS    0X                    END OF ISTSLD
```

Restore Data Structure (ISTSREST)

```
  LOC     SOURCE STATEMENT
000000 ISTSREST DSECT                       RESTORE STRUCTURE
       *
       *
       *
000000 SRENAME    DS  CL8                   LU NAME
000008 SREMODE    DS  CL8                   LU MODE
000010 SRENXTAD   DS  A                     NEXT RESTORE STRUCTURE ADDRESS
000014 SRESLDAD   DS  A                     SLD STRUCTURE ADDRESS
000018 SRESESAD   DS  A                     ADDRESS OF FIRST SRESESS
00001C SREMFLGS   DS  XL2                   MODE LEVEL FLAGS
       SREMDRS    EQU X'80'                 1=MODE HAS BEEN RESTORED
00001E SRESESCT   DS  H                     NUMBER OF SRESESS STRUCTURES
000020 SRENETID   DS  CL8                   NETID OF LU
000028 SREEND     DS  0X                    END OF ISTSREST STRUCTURE
       *
000000 SRESESS    DSECT                     SESSION INFORMATION
000000 SRESNXTA   DS  A                     NEXT SESSION STRUCTURE ADDRESS
000004 SRESFLGS   DS  XL3                   SESSION LEVEL FLAGS
       SREPCONV   EQU X'80'                 1=CONVERSATION PENDING DEALLOCATION
       *                                    FOR PERSISTENT LU-LU SESSIONS
       SRESPNDA   EQU X'40'                 1=SESSION PENDING DEACTIVATION
       *                                    FOR PERSISTENT LU-LU SESSIONS
000007 SRESIDL    DS  XL1                   SESSION INSTANCE IDENTIFIER LENGTH
000008 SRESESID   DS  XL8                   SESSION INSTANCE IDENTIFIER
000010 SRESEND    DS  0X                    END OF SESSION INFORMATION
```

Status Data Structure (ISTSTATD)

```
  LOC     SOURCE STATEMENT
000000 ISTSTATD DSECT                       TESTSTAT INFORMATION ENTRY
       *
000000 STATENTL DS    XL2                   LENGTH OF THIS ENTRY
000002 STATENTT DS    X                     ENTRY TYPE
       STATNRME EQU   X'01'                 NORMAL DATA INFORMATION ENTRY
       STATXPDE EQU   X'02'                 EXPEDITED DATA INFORMATION ENTRY
       STATRTSE EQU   X'03'                 REQUEST-TO-SEND INFORMATION ENTRY
000003 STAFLAG1 DS    X                     STATUS ENTRY FLAGS
       STACNVCA EQU   X'80'                 DATA IS IN CA MODE
000004 STACNVID DS    XL4                   CONVID OF CONVERSATION
000008 STATOTAV DS    XL4                   TOTAL DATA AVAILABLE (NORM & EXPD)
00000C STACURLL DS    XL2                   CURRENTLY ACTIVE LL FIELD (NORM),
       *                                    RESERVED (EXPED & RTS_RCVD)
00000E STACURLR DS    XL2                   CURRENT LL REMAINDER (NORM),
       *                                    RESERVED (EXPED & RTS_RCVD)
000010 STATENTE DS    0X                    END OF STATUS ENTRY
```

Feedback Code Data Structure (ISTUSFBC)

```
  LOC     SOURCE STATEMENT
000000 ISTUSFBC DSECT
       *
       ***********************************************************************
       *                                                                     *
       *        THE FOLLOWING CODES ARE STORED IN EITHER 'RPLRTNCD', OR    *
       *        'RPLFDB2'' OR ''RPLFDB3''.  SEE THE INTRODUCTORY COMMENTS   *
       *        FOR EACH GROUP FOR FURTHER INFORMATION.                     *
       *                                                                    *
       *              RPL FIELD NAME           OPERAND OF MANIPULATIVE MACRO *
       *                                                                    *
       *              RPLRTNCD                 RTNCD     (FEEDBACK CODE)     *
       *              RPLFDB2                  FDBK2     (REASON CODE)       *
       *              RPLFDB3                  FDBK      (DATA FLAGS)        *
       *                                                                    *
       *        IF THE RPLRTNCD IS SET TO X'00' AND THE RPLFDB2 IS SET      *
       *        TO X'1A' THEN THE USER SHOULD REFER TO THE FOLLOWING FIELDS  *
       *        IN THE RPL6.  THIS IS ADDED FOR APPC/VTAM.                  *
       *                                                                    *
       *              RPL6 FIELD NAME                                       *
       *                                                                    *
       *              RPL6RCPR                 PRIMARY RETURN CODE          *
       *              RPL6RCSC                 SECONDARY RETURN CODE        *
       *                                                                    *
       ***********************************************************************
       *
       *
       *****    RPLRTNCD        CONTAINS A FEEDBACK CODE.  IF THE RPL    *****
       *                        REQUEST IS UNSUCCESSFUL THEN REGISTER        *
       *                        ZERO WILL ALSO CONTAIN THIS CODE. FOR A      *
       *                        CERTAIN GROUP OF ERRORS, ONLY REGISTER       *
       *                        ZERO WILL CONTAIN THE FEEDBACK CODE AND      *
       *                        NO FEEDBACK INFORMATION WILL BE PLACED IN    *
       *                        THE RPL.                                     *
       *                                                                     *
       *                        THE FEEDBACK CODE EQUATES ARE AS FOLLOWS:    *
       USFAOK   EQU   X'00'          NORMAL COMPLETION/CONDITIONAL COMPLETION
       USFXORDC EQU   X'04'          EXTRAORDINARY COMPLETION
       USFRESSU EQU   X'08'          REISSUE THIS REQUEST
       USFDAMGE EQU   X'0C'          DAMAGE - INTEGRITY OF REQUEST/DEVICE
       USFENVER EQU   X'10'          ENVIRONMENT ERROR
       USFLOGIC EQU   X'14'          USER LOGIC ERROR
       USFRLGIC EQU   X'18'          USER LOGIC ERROR - SETONLY IN REG ZERO
       USF6CHEK EQU   X'20'          RPL/RPL6 IN WRONG STATE - SET ONLY IN
       *                               REG00
       USF6WRCK EQU   X'24'          WRONG CHECK MACRO ISSUED - SET ONLY IN
       *                               REG00
       *
       *
       ***********************************************************************
       *                                                                     *
       *        RPLFDB2         CONTAINS A REASON CODE.  THIS REASON CODE    *
       *                        INDICATES ADDITIONAL INFORMATION ABOUT THE   *
       *                        FEEDBACK CODE.                               *
       *                                                                     *
       *****    REASON CODE EQUATES FOR RPLFDB2 IF RPLRTNCD EQUALS X'00' *****
       *                                                                     *
       USFAOOK  EQU   X'00'          OPERATION SUCCESSFULLY COMPLETED
       USFRCWNP EQU   X'01'          RESET CONDITIONAL WAS NO-OPED
       USFRCDPR EQU   X'02'          RESET CONDITIONAL SUCCESSFUL -
       *                             READ-AHEAD DATA PRESENT
       USFYTCTN EQU   X'03'          YIELDED TO CONTENTION
       USFYTCTL EQU   X'04'          YIELDED TO CONTENTION, ERROR LOCK SET
       USFATSFI EQU   X'05'          AREA TOO SMALL FOR INQUIRE/INTERPRET
       USFNOIN  EQU   X'06'          NO INPUT AVAILABLE
```

```
USFIIINA EQU   X'07'          INQUIRE INFORMATION NOT AVAILABLE
USFDSTIU EQU   X'08'          DESTINATION IN USE
USFNLGFA EQU   X'09'          NO LOGON FOUND FOR ACCEPT MATCH
USFANC   EQU   X'0A'
USF6APPC EQU   X'0B'          INDICATES THAT AN ERROR OCCURRED RUNNING
*                             APPC, AND REFER TO THE RPL6 PRIMARY AND
*                             SECONDARY RETURN CODES
USFINQPS EQU   X'0D'          MORE SESSIONS PENDING RECOVERY ON
*                             WHICH TO INQUIRE
*                                                                     *
*        IF, FOLLOWING A SYNCHRONOUS RPL REQUEST MACRO OR CHECK       *
*        MACRO, REGISTER 15 CONTAINS X'00' THEN REGISTER ZERO WILL    *
*        CONTAIN ONE OF THE ABOVE REASON CODE VALUES                  *
*                                                                     *
*
*****    REASON CODE EQUATES FOR RPLFDB2 IF RPLRTNCD EQUALS X'04' *****
*
USFRVIRC EQU   X'00'          RVI RECEIVED, ERROR LOCK SET
USFATNRC EQU   X'01'          ATTENTION RECEIVED, ERROR LOCK SET
USFBSCSM EQU   X'02'          BSC STATUS MSG RECEIVED
USFEXRQ  EQU   X'03'          EXCEPTION REQUEST RECEIVED
USFEXRS  EQU   X'04'          EXCEPTION RESPONSE RECEIVED
USFNQN   EQU   X'05'          RESOURCE KNOWN AS NQN ONLY
*
*
*****    REASON CODE EQUATES FOR RPLFDB2 IF RPLRTNCD EQUALS X'08' *****
*
USFSTALF EQU   X'00'          TEMPORARY OUT OF STORAGE SITUATION EXISTS
*                             RPL ECB/EXIT NOT POSTED/INVOKED         *
*
*
*****    REASON CODE EQUATES FOR RPLFDB2 IF RPLRTNCD EQUALS X'0C' *****
*
USFIOEDU EQU   X'00'          I/O ERROR, DEVICE STILL USABLE ER LK SET
USFDVUNS EQU   X'01'          I/O ERROR, DEVICE NOT USABLE ER LCK SET
USFUNTRM EQU   X'02'          REQUEST RESET BY TEST REQUEST MESSAGE
USFBTHEX EQU   X'03'          BUFFER THRESHOLD EXCEEDED
USFBTEOR EQU   X'04'          BUF THRESHOLD EXCEEDED, ONLY READS ALLOW
USFNCPAO EQU   X'05'          NCP ABENDED, RESTART O.K.
USFLIORP EQU   X'06'          LAST I/O REQUEST PURGED
USFRECIP EQU   X'07'          RECOVERY IN PROGRESS
USFRTRAF EQU   X'08'          RECORD TERMINAL RESTARTED AFTER FAILURE
USFQOPDC EQU   X'09'          QUEUED OPNDST CANCELLED BY CLSDST
USFUSRES EQU   X'0A'          REQUEST RESET BY THE USER
USFCLOCC EQU   X'0B'          CLSDST OR TERMSESS ISSUED OR UNBIND SENT
*                             IN LIEU OF NEGATIVE BIND RESPONSE
*
USFCLRED EQU   X'0C'          REQUEST WAS CLEAR'ED
USFPREXC EQU   X'0D'          SEND CANCELLED DUE PRIOR EXCEPTION COND.
USFPOQLE EQU   X'0E'          SEND CANCELLED DUE POA QUEUE LIMIT
*
*
*
*****    REASON CODE EQUATES FOR RPLFDB2 IF RPLRTNCD EQUALS X'10' *****
*
USFTANAV EQU   X'00'          TERMINAL OR APPLICATION NOT AVAILABLE
USFSBFAL EQU   X'01'          SESSION BIND FAILED
USFTAPUA EQU   X'02'          TARGET APPLICATION UNACCEPTABLE
USFVTHAL EQU   X'03'          VTAM IS HALTING
USFILRS  EQU   X'04'          INCOMPATIBLE DEFINITION
USFPCF   EQU   X'05'          PERMANENT FAILURE IN PATH
USFANS   EQU   X'06'          AUTO NETWORK SHUTDOWN
USFVOFOC EQU   X'07'          VARY DEACTIVATE IMMEDIATE OCCURRED
USFDISCO EQU   X'08'          DISCONNECT OCCURRED
USFUTSCR EQU   X'09'          UNCONDITIONAL TERMINATE SELF CMD RECEIVED
USFSYERR EQU   X'0A'          APPARENT VTAM ERROR
USFDIDOL EQU   X'0B'          DISCONNECT ON DIAL-OUT LINE
```

```
USFDIDIL EQU   X'0C'          DISCONNECT ON DIAL-IN LINE
*    NOTE - X'0D' AND X'0E' - RPL ECB/EXIT NOT POSTED/INVOKED        *
USFVTMNA EQU   X'0D'          VTAM INACTIVE FOR THAT APPLICATION
USFABNDO EQU   X'0E'          ABEND CONDITION HAS OCCURRED
*                                                                     *
USFVTBFO EQU   X'0F'          VTAM BUFFER OVERFLOW
USFCTERM EQU   X'10'          CONDITIONAL TERM SELF
USFOSDTF EQU   X'11'          SDT FAILURE ON OPNDST
USFMFF   EQU   X'12'          MACRO FUNCTION FAILED,SENSE INCLUDED
USF6APRJ EQU   X'13'          ATTEMPT TO START 6.2 SESSION: REQUEST
*                              REJECTED
USF6APST EQU   X'14'          ATTEMPT TO START 6.2 SESSION: PENDING
*                              SESSION TERMINATED
USF6APIS EQU   X'15'          MUST ISSUE APPCCMD
USFNONSW EQU   X'16'          SWITCHED OPERATION ATTEMPTED ON
*                              NONSWITCHED DEVICE
USFNOCRY EQU   X'17'          ENCRYPTION REQUESTED WHEN SESSION
*                             DOES NOT SUPPORT CRYPTOGRAPHY
USFNOSES EQU   X'18'          XES IS NOT ACCESSABLE
USFNOSYS EQU   X'19'          APPLICATION NOT RESIDENT IN A SYSPLEX
*
USFXMEMS EQU   X'1A'          SUSPEND FAILURE
USFXMEMR EQU   X'1B'          RESUME FAILURE
USFOSLVL EQU   X'1C'          OPERATING SYSTEM LEVEL INSTALLED DOES NOT
*                              SUPPORT THE REQUESTED FUNCTION
USFSECME EQU   X'1D'           SECURITY MANAGER ERROR
*
*
*****    REASON CODE EQUATES FOR RPLFDB2 IF RPLRTNCD EQUALS X'14' *****
*
USFNONVR EQU   X'00'          RPL CONTAINS A NON-VTAM REQUEST CODE
*                             RPL ECB/EXIT NOT POSTED/INVOKED         *
USFNOTAS EQU   X'01'     NOT ASSIGNED
USFEXTAZ EQU   X'02'          RPL INDICATES EXIT, EXIT ADDR IS ZERO
*                             RPL ECB/EXIT NOT POSTED/INVOKED         *
USFEXTEZ EQU   X'03'          RPL IND EXTERNAL ECB, ECB ADDR IS ZERO
*                             RPL ECB/EXIT NOT POSTED/INVOKED         *
USFCRPLN EQU   X'04'          CHECKED RPL IS NOT ACTIVE
*                        ONLY OCCURS FOLLOWING A CHECK MACRO REQUEST  *
USFCBERR EQU   X'10'          RPL POINTS TO INVALID ACB
USFRNORT EQU   X'11'          NO RTYPE SPECIFIED
USFCLSIP EQU   X'12'          CLSDST IN PROGRESS
USFCIDNG EQU   X'13'          CID IS INVALID
USFILDOP EQU   X'14'          LDO COMMAND FIELD IS INVALID
USFWANCR EQU   X'15'          READ NOT CHAINED
USFSTOOD EQU   X'16'          SOLICIT SPECIFIC TO OUTPUT ONLY DEVICE
USFRTOOD EQU   X'17'          READ TO OUTPUT ONLY DEVICE
USFWTOI  EQU   X'18'          WRITE TO INPUT ONLY DEVICE
USFEWNS  EQU   X'19'          ERASE TO INVALID DEVICE
USFEWAU3 EQU   X'1A'          WRITE EAU TO NON-3270
USFCWTOO EQU   X'1B'          WRITE CONV TO OUTPUT ONLY
USFCWB   EQU   X'1C'          WRITE WITH ERASE AND CONV SPECIFIED
USFCCCPY EQU   X'1D'          CHAINED COPY LDO
USFIDA   EQU   X'1E'          INVALID DATA AREA OR LENGTH
USFILDOA EQU   X'1F'          INVALID LDO ADDRESS
USFJTOJ  EQU   X'20'          JUMP LDO TO JUMP
USFMT100 EQU   X'21'          MORE THAN 100 LDOS
USFRILCP EQU   X'22'          RESET LDO IS NOT ALONE
USFCRIRT EQU   X'23'          INVALID MACRO REQUEST TYPE
USFASIDE EQU   X'24'          ASID MISMATCH
USFEWBLK EQU   X'25'          WRITE ERASE BLOCK
USFCRSDC EQU   X'26'          SOLICIT LDO WITH DATA CHAINING
USFIREST EQU   X'27'          RESET OPTION CODE INVALID
USFWBT32 EQU   X'28'          WRITE BLOCK TO 3270 DEVICE
USFRMD32 EQU   X'29'          READ MODIFIED TO NON-3270 DEVICE
USFCTN32 EQU   X'2A'          COPY TO NON-3270 DEVICE
USFWCNVR EQU   X'2B'          WRITE CONV ISSUED WHEN DATA EXPECTED
```

```
USFRNFT3 EQU   X'2C'          OUTPUT NOT PRECEDED BY INPUT
USFRCINV EQU   X'2D'          RESET CONDITIONAL ILLEGAL
USFINVRM EQU   X'2E'          INVALID READ MODE
USFLGCNT EQU   X'2F'          EXCESSIVE LEADING GRAPHICS, ERROR LK SET
USFCPCNT EQU   X'30'          COPY COUNT ERROR
USFIDAEL EQU   X'31'          INVALID DATA AREA OR LENGTH, ERROR LK SET
USFUSELE EQU   X'32'          REQUEST INVALID FOR DEVICE, ERROR LK SET
USFCRNF  EQU   X'33'          CONV. REPLY NOT POSSIBLE, ERROR LOCK SET
USFNORD  EQU   X'34'          NO READ WHERE REQUIRED, ERROR LOCK SET
USFCPYE2 EQU   X'35'          COPY WRONG CLUSTER, ERROR LOCK SET
USFRELNP EQU   X'36'          REQUEST LOCK NOT ALLOWED, ERROR LOCK SET
USFCPYE1 EQU   X'37'          COPY UNOPENED DEVICE, ERROR LOCK SET
USFDFIBH EQU   X'38'          FIRST I/O FAILED INVALID BHSET, ER LK SET
USFDFIPO EQU   X'39'          FIRST I/O FAILED INVALID PROC, ER LK SET
USFQSCIE EQU   X'3A'          QUIESCE IN EFFECT
USFREXAL EQU   X'3B'          RESPOND = EX ALONE IN RPL
USFSDNP  EQU   X'3C'          POST = SCHED STILL OUTSTANDING
USFSCEM  EQU   X'3D'          CHAINING ERROR: MIDDLE OR LAST REQUIRED
USFSCEF  EQU   X'3E'          CHAINING ERROR: FIRST OR ONLY REQUIRED
USFSNQC  EQU   X'3F'          QUIESCE COMPLETE RESPONSE NOT REQUESTED
USFSINVC EQU   X'40'          INVALID CONTROL = OPTION
USFSDFR  EQU   X'41'          NO START DATA TRAFFIC IN EFFECT
USFSNOS  EQU   X'42'          CONTROL RESPONSE INVALID
USFSNOUT EQU   X'43'          SEND RESPONSE NOT REQUESTED
USFLIMEX EQU   X'44'          NIB RESPLIM EXCEEDED
USFSSEQ  EQU   X'45'          SEQUENCE NUMBER ERROR
USFSINVS EQU   X'46'          RESPOND = OPTION MISMATCH
USFSINVR EQU   X'47'          RESP = OPTION INVALID FOR POST = RESP
USFINVRT EQU   X'48'          PROTOCOL VIOLATION
USFACINV EQU   X'49'          INVALID ACTION TYPE
USFICNDN EQU   X'4A'          INSTALLATION EXIT ROUTINE N/A
USFILSIN EQU   X'4B'          INVALID LOGON SEQUENCE
USFIICBE EQU   X'4C'          LU NOT SESSION CAPABLE
USFINTNA EQU   X'4D'          NO INTERPRET TABLE
USFILNBL EQU   X'4E'          ILLEGAL USE OF NIB LIST
USFINVOT EQU   X'4F'          INVALID OPNDST TYPE
USFINVAP EQU   X'50'          INVALID AQUIRE PARAMETER
USFAPNAC EQU   X'51'          APPLICATION NEVER ACCEPTS
USFINVNB EQU   X'52'          INVALID NIB
USFSYMNU EQU   X'53'          SYMBOLIC NAME UNKNOWN
USFDSTUO EQU   X'54'          DESTINATION UNOPENABLE
USFNOPAU EQU   X'55'          NO OPNDST AUTHORIZATION
USFMDINC EQU   X'56'          MODE - DEVICE INCOMPAT
USFINVMD EQU   X'57'          INVALID MODE
USFBHSUN EQU   X'58'          BHSET NAME UNKNOWN
USFMDNAU EQU   X'59'          MODE NAME AUTHORIZED
USFMBHSS EQU   X'5A'          MULTIPLE BHSETS SPECIFIED
USFINVLA EQU   X'5B'          INVALID LOGON DATA AREA
USFDUPND EQU   X'5C'          DUPLICATE NODES
USFDSTNO EQU   X'5D'          DESTINATION NOT OPENED
USFNPSAU EQU   X'5E'          NO PASS AUTHORIZATION
USFRSCNO EQU   X'5F'          RESOURCE NOT OWNED
USFRSCNC EQU   X'60'          RESOURCE NOT CLOSEABLE
USFINVSL EQU   X'61'          INVALID SETLOGON
USFMCNVD EQU   X'62'          MACRO NOT VALID FOR SPECIFIED DEVICE
USFRNOEL EQU   X'6C'          PROGRAM OPERATOR APPLICATION EXCEEDED
*                             LIMIT OF OUTSTANDING RCVCMDS
USFRNONA EQU   X'6D'          APPLICATION NOT AUTHORIZED
USFRNOSE EQU   X'6E'          REPLY, SENT BY PROGRAM OPERATOR,
*                             REJECTED DUE TO SYNTAX ERROR
USFRNOIA EQU   X'6F'          PROGRAM OPERATOR INTERFACE INACTIVE
USFRNOCL EQU   X'70'          RCVCMD REJECTED BECAUSE PROGRAM
*                             OPERATOR APPLICATION IS CLOSING
USFRNOCE EQU   X'71'          V,D,F, SENT BY PROGRAM OPERATOR
*                             REJECTED DUE TO SYNTAX ERROR
USFPCIT  EQU   X'72'          LOGICAL ERROR, PRIMARY CANNOT ISSUE
*                             TERMSESS
```

```
USFINVSD EQU   X'73'          INVALID OPTIONS ON SEND
*                                                                       *
USFNRNBD EQU   X'74'          NEGOTIABLE RESPONSE TO NON-NEGOTIABLE
*                             BIND                                      *
USFINBRP EQU   X'75'          INVALID NEGOTIABLE BIND RESPONSE
*                             PARAMETERS                                *
USFINBSZ EQU   X'76'          INVALID NEGOTIABLE BIND RESPONSE
*                             SIZE                                      *
USFNFMDQ EQU   X'77'          FM DATA REQUEST UNIT
*                             REQUIRED
USFCHINV EQU   X'78'          INVALID CHAIN
*                             SPECIFICATION
USFBLINV EQU   X'79'          INVALID BUFFER LIST
*                             LENGTH
USFINVRH EQU   X'7B'          INVALID USER
*                             RH
USFSCINV EQU   X'7C'          OPTCD=USERRH INVALID FOR
*                             SESSIONC
USFHPINV EQU   X'7D'          XRF PROTOCOL VIOLATION
USFCOMR  EQU   X'7E'          CONFLICTING OPTCD ON A MACRO REQUEST
*
USF6PENA EQU   X'7F'          POLICING ERROR - NON-APPC MACRO
USFPRINV EQU   X'80'          PERSISTENT LU-LU SESSION SUPPORT
*                             REQUESTED FOR APPLICATION THAT IS NOT
*                             PERSISTENT SESSION CAPABLE
USFTSPND EQU   X'81'          TERMSESS WITHOUT UNBIND WITH SESSION IN
*                             PENDING ACTIVE STATE
USFPARML EQU   X'82'          PARAMETER LENGTH INVALID
USFSFERR EQU   X'83'          SUBFIELD NOT SUPPORTED, INVALID
*                             COMBINATION OF SUBFIELDS, OR
*                             SUBFIELD FORMAT ERROR
USFASDAZ EQU   X'84'          ZERO NIBASDPA FIELD
USFSMBRS EQU   X'85'          SESSION IS IN RECOVERY STATE AND MUST BE
*                             RESTORED
USFSESSA EQU   X'86'          SESSIONS OR AFFINITIES EXIST
USFSNAME EQU   X'87'          RESOURCE NAME AND GENERIC NAME EQUAL
USFNOSPT EQU   X'88'          NO SPT EXISTS
USFNSECM EQU   X'89'          NO SECURITY AUTHORIZATION FOR GENERIC
*                             RESOURCE
USFDIFNM EQU   X'8A'          ALREADY REGISTERED WITH A DIFFERENT
*                             GENERIC NAME
USFNOMAP EQU   X'8B'          NOT REGISTERED AS A GENERIC RESOURCE
USFNETID EQU   X'8C'          ALREADY REGISTERED WITH A DIFFERENT
*                             NETWORK ID
USFCPNAM EQU   X'8D'          MAPPING ALREADY EXISTS ON A DIFFERENT
*                             SYSPLEX NODE
USFCNFAC EQU   X'8E'          CONFLICTING APPC CAPABILITY
USFVTAMO EQU   X'8F'          SPTE IS OWNED BY VTAM
USFUSVAR EQU   X'90'          GENERIC NAME CONFLICTS WITH AN
*                             EXISTING USERVAR
USFGNUNA EQU   X'91'          TSO GENERIC NAME CONFLICT
USFGGMNA EQU   X'92'          SETLOGON GNAMESUB FAILURE
USFSTKNV EQU   X'93'          STOKEN NOT VALID
*
*****    NO REASON CODE EQUATES EXIST FOR RPLRTNCD EQUALS X'18'   *****
*                                                                       *
*
*****    EQUATES FOR RPLFDB3 ON RETURN FROM INQUIRE IF            *****
*****    RPLRTNCD IS X'00'                                        *****
*
USFIACT  EQU   X'00'          APPLICATION IS ACTIVE
USFIINA  EQU   X'04'          APPLICATION IS INACTIVE
*                             SEE USFANC (X'0A') UNDER RPLFDB2
*                             WHEN RTNCD = X'00'
USFINA   EQU   X'08'          APPLICATION WILL NOT ACCEPT LOGONS
USFITNA  EQU   X'0C'          APPLICATION IS TEMPORARILY NOT
*                             ACCEPTING LOGONS
```

```
USFIQUIE EQU   X'10'          APPLICATION IS QUIESCING
USFILACT EQU   X'80'          RESOURCE IS ACTIVE
USFILINA EQU   X'84'          RESOURCE IS NOT ACTIVE
*
*
**********************************************************************
***
*
***    THE FOLLOWING ARE ALL THE RPL6RCPR (PRIMARY RETURN
***    CODE) VALUES FOR APPC/VTAM.
***
USF6OK   EQU   X'0000'        OK
USF6ALLC EQU   X'0004'        ALLOCATION ERROR
USF6CNSA EQU   X'0008'        CNOS ALLOCATION ERROR
USF6CNSN EQU   X'000C'        CNOS RESOURCE FAILURE, NO RETRY
USF6CRRJ EQU   X'0010'        COMMAND RACE REJECT
USF6DABP EQU   X'0014'        DEALLOCATE ABEND PROGRAM
USF6DABS EQU   X'0018'        DEALLOCATE ABEND SERVICE
USF6DABT EQU   X'001C'        DEALLOCATE ABEND TIMER
USF6CNSR EQU   X'0020'        CNOS FAILURE, RETRY
USF6LRBE EQU   X'0024'        LOGICAL RECORD BOUNDARY ERROR
USF6SLCL EQU   X'0028'        LU MODE SESSION LIMIT CLOSED
USF6PARM EQU   X'002C'        PARAMETER ERROR
USF6PENT EQU   X'0030'        PROGRAM ERROR NO TRUNCATION
USF6PEPU EQU   X'0034'        PROGRAM ERROR PURGING
USF6PETR EQU   X'0038'        PROGRAM ERROR TRUNCATING
USF6SENT EQU   X'003C'        SERVICE ERROR NO TRUNCATION
USF6SEPU EQU   X'0040'        SERVICE ERROR PURGING
USF6SETR EQU   X'0044'        SERVICE ERROR TRUNCATING
USF6RFNR EQU   X'0048'        RESOURCE FAILURE, NO RETRY
USF6RFRE EQU   X'004C'        RESOURCE FAILURE, RETRY
USF6STER EQU   X'0050'        STATE ERROR
USF6URMD EQU   X'0054'        UNRECOGNIZED MODE NAME
USF6UNSC EQU   X'0058'        UNSUCCESSFUL, SESSION NOT AVAILABLE
*
USF6UECR EQU   X'005C'        USER ERROR CODE RECEIVED
USF6NOFM EQU   X'0060'        NO FMH5 AVAILABLE
USF6ACFL EQU   X'0064'        ACTIVATION FAILURE
USF6SLEX EQU   X'0068'        LU MODE SESSION LIMIT EXCEEDED
USF6SACT EQU   X'006C'        SESSION NOT PENDING
USF6STOR EQU   X'0070'        TEMPORARY STORAGE SHORTAGE OR RESOURCE
*                             SHORTAGE
USF6HALT EQU   X'0074'        HALT ISSUED
USF6VIYA EQU   X'0078'        VTAM INACTIVE FOR YOUR ACB
USF6RQAB EQU   X'007C'        REQUEST ABORTED
USF6DLNR EQU   X'0080'        DEALLOCATE NORMAL
USF6STSH EQU   X'0084'        STORAGE SHORTAGE
USF6CREJ EQU   X'0088'        CANCELLED BY REJECT OR DEALLOCATE ABND*
*
USF6PROE EQU   X'008C'        PARTNER COMMITTED PROTOCOL VIOLATION
*
USF6NOTA EQU   X'0090'        APPLICATION NOT APPC CAPABLE
USF6SDRJ EQU   X'0094'        SEND DATA REJECTED INVALID STATE
USF6STGS EQU   X'0098'        STORAGE SHORTAGE WHILE SENDING
*                             DATA
USF6RSTF EQU   X'009C'        RESTORE REJECTED
USF6RNAL EQU   X'00A0'        REQUEST NOT ALLOWED
USF6SPMD EQU   X'00A4'        MODE MUST BE RESTORED BEFORE USING
USF6ENVE EQU   X'00A8'        ENVIRONMENT ERROR
USF6ERIN EQU   X'00AC'        ERROR INDICATION WAS RECEIVED
USF6NRER EQU   X'00B0'        NAME RESOLUTION ERROR
USF6CSME EQU   X'00B4'        CSM DETECTED ERROR
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'0000' (USF6OK).
***
```

```
USF6OKSC EQU   X'0000'        OK
USF6ASSP EQU   X'0001'        AS SPECIFIED
USF6ASNG EQU   X'0002'        AS NEGOTIATED
USF6RCVR EQU   X'0003'        RECEIVE SPECIFIC REJECTED
USF6SNGL EQU   X'0004'        PARTNER LU SUPPORTS SINGLE SESSION
*
USF6INER EQU   X'0005'        INTERNAL VTAM ERROR
USF6RSUN EQU   X'0006'        RESTORE UNNECESSARY - NO SESSIONS
*                             TO RESTORE
USF6RSIN EQU   X'0007'        RESTORE INCOMPLETE - INPUT WORK
*                             AREA TOO SMALL
USF6NINA EQU   X'0008'        NO IMMEDIATELY AVAILABLE INFORMATION
*                             FOR REQUEST
USF6RTEC EQU   X'0009'        REQUEST TERMINATED BY END OF
*                             CONVERSATION
USF6ANMS EQU   X'000A'        SESSIONS WILL USE APPL NETWORK NAME,
*                             GENERIC NAME WAS REQUESTED
USF6GNMS EQU   X'000B'        SESSIONS WILL USE GENERIC NAME,
*                             APPL NETWORK NAME WAS REQUESTED
USF6NAM1 EQU   X'000C'        AS SPECIFIED, PARTNER LU KNOWN BY
*                             DIFFERENT NAME
USF6NAM2 EQU   X'000D'        AS NEGOTIATED, PARTNER LU KNOWN BY
*                             DIFFERENT NAME
*
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'0004' (USF6ALLC).
***
USF6ALNR EQU   X'0000'        ALLOCATION FAILURE, NO RETRY
USF6ALR  EQU   X'0001'        ALLOCATION FAILURE, RETRY
USF6ALCM EQU   X'0002'        CONVERSATION TYPE MISMATCH
USF6ALPI EQU   X'0003'        PIP NOT ALLOWED
USF6ALPP EQU   X'0004'        PIP NOT SPECIFIED CORRECTLY
USF6ALSC EQU   X'0005'        SECURITY NOT VALID
USF6ALSY EQU   X'0006'        SYNC LEVEL NOT SUPPORTED BY LU
USF6ALSL EQU   X'0007'        SYNC LEVEL NOT SUPPORTED BY PROGRAM
*
USF6ALTP EQU   X'0008'        TPN NOT RECOGNIZED
USF6ALTN EQU   X'0009'        TRANSACTION PROGRAM NOT AVAILABLE, NO
*                             RETRY
USF6ALTR EQU   X'000A'        TRANSACTION PROGRAM NOT AVAILABLE, RETRY
*
USF6ALRN EQU   X'000B'        CANNOT RECONNECT TRANSACTION PROGRAM,
*                             NO RETRY
USF6ALRR EQU   X'000C'        CANNOT RECONNECT TRANSACTION PROGRAM,
*                             RETRY
USF6ALNS EQU   X'000D'        RECONNECT NOT SUPPORTED BY PROGRAM
*
USF6SPMA EQU   X'000E'        MODE MUST BE RESTORED BEFORE USING
USF6DARQ EQU   X'000F'        DEALLOCATION REQUESTED
USF6ALSF EQU   X'0010'        REQUESTED SYNCH LEVEL NOT ALLOWED
*                             FOR FULL-DUPLEX CONVERSATION
USF6LNSF EQU   X'0011'        LU PAIR NOT SUPPORTING FULL-DUPLEX
*                             CONVERSATIONS
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'0008' (USF6CNSA).
***
USF6CANR EQU   X'0000'        ALLOCATION FAILURE, NO RETRY
USF6CAR  EQU   X'0001'        ALLOCATION FAILURE, RETRY
USF6CATR EQU   X'0002'        TRANSACTION PROGRAM NOT AVAILABLE, RETRY
*
USF6CATN EQU   X'0003'        TRANSACTION PROGRAM NOT AVAILABLE, NO
*                             RETRY
USF6CACM EQU   X'0004'        CONVERSATION TYPE MISMATCH
```

```
USF6CASC EQU   X'0005'        SECURITY NOT VALID
USF6SPMC EQU   X'0006'        MODE MUST BE RESTORED BEFORE USING
USF6NQNM EQU   X'0007'        NETWORK QUALIFIED NAME MISMATCH
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'10' (USF6CRRJ).
***
USF6CRPR EQU   X'0000'        PARTNER GRANTED RETRY
USF6CRLR EQU   X'0001'        CONTROL OPERATOR OF LOCAL LU RETRIED
*
USF6PCIP EQU   X'0002'        PARTNER CNOS IN PROGRESS
USF6LPSS EQU   X'0003'        LU IS IN PENDING SINGLE STATE
USF6PLSS EQU   X'0004'        PARTNER LU STARTING SESSION
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'002C' (USF6PARM).
***
USF6IVLU EQU   X'0000'        INVALID LU NAME OR NETID
USF6IVMD EQU   X'0001'        INVALID MODE
USF6IVCI EQU   X'0002'        INVALID CONVERSATION ID
USF6IVLL EQU   X'0003'        INVALID LL
USF6IVSV EQU   X'0004'        INVALID VALUES FOR SNASVCMG MODE
USF6IVDL EQU   X'0005'        INVALID DRAINL CHANGE
USF6SNAR EQU   X'0006'        SNASVCMG MODE CANNOT CURRENTLY BE RESET
*
USF6MMEX EQU   X'0007'        MINWINL PLUS MINWINR EXCEEDS SESSLIM
*
USF6LNIN EQU   X'0008'        SUPPLIED LENGTH INSUFFICIENT
USF6INSL EQU   X'0009'        INCOMPLETE SESSION LIMITS STRUCTURE
*                             SUPPLIED
USF6INFM EQU   X'000A'        INCOMPLETE FMH5 SUPPLIED
USF6INGD EQU   X'000B'        INCOMPLETE GDS VARIABLE SUPPLIED
*
USF60EXT EQU   X'000C'        ZERO EXIT FIELD
USF60ECB EQU   X'000D'        ZERO ECB FIELD
USF6RIAS EQU   X'000E'        REQUEST INVALID FOR ADDRESS SPACE
*
USF6CBIN EQU   X'000F'        CONTROL BLOCK INVALID
USF6INDL EQU   X'0010'        INVALID DATA ADDRESS OR LENGTH
USF6PRVO EQU   X'0011'        PREVIOUS MACRO INSTRUCTION OUTSTANDING
*
USF6BLIV EQU   X'0012'        BUFFER LIST LENGTH INVALID
USF6NOMD EQU   X'0013'        NO CORRESPONDING MODE IN LM TABLE
*
USF6IVBP EQU   X'0014'        INVALID BIND PARAMETERS
USF6IVTP EQU   X'0015'        INVALID TPN
USF6NOLU EQU   X'0016'        NO CORRESPONDING LU IN LM TABLE
*
USF6IMDF EQU   X'0017'        INVALID MODE SPECIFIED
USF6ILSP EQU   X'0018'        INVALID LIMIT SPECIFIED
USF6SMAI EQU   X'0019'        SNASVCMG MODE ALREADY INITIALIZED
*
USF6ALLS EQU   X'001A'        ALL MODES SPECIFIED ON SINGLE SESSION LU
*
USF6SMSS EQU   X'001B'        SNASVCMG MODE FOR SINGLE SESSION LU
*
USF6SSMI EQU   X'001C'        SINGLE SESSION, MODE ALREADY INITIALIZED
*
USF6CIDI EQU   X'001E'        CID INVALID
USF6APNA EQU   X'001F'        APPCCMD ISSUED FOR NON-APPC
USF6PRRO EQU   X'0020'        PREVIOUS REJECT REQUEST OUTSTANDING
*
USF6DARJ EQU   X'0021'        DEALLOCATE ABND* REJECTED, RETRY
USF6IVCQ EQU   X'0022'        INVALID CONTROL OR QUALIFY VALUE
USF6INSI EQU   X'0023'        INVALID SESSION INSTANCE IDENTIFIER
USF6PSHI EQU   X'0024'        PS HEADER NOT SUPPLIED
USF6PSLI EQU   X'0025'        PS HEADER LENGTH INSUFFICIENT
```

```
USF6NMSC EQU   X'0026'        SESSION INSTANCE IDENTIFIER AND
*                             CONVERSATION ID MISMATCH
USF6IDET EQU   X'0027'        INVALID DEACTIVATION TYPE CODE
USF6NCRY EQU   X'0028'        CRYPTOGRAPHY NOT ALLOWED ON MODE
USF6INLI EQU   X'0029'        INVALID LIST VALUE SPECIFIED ON
*                             APPCCMD FOR RESTORE
USF6INCG EQU   X'002A'        INVALID CGID VALUE ON ALLOCATE
USF6NONI EQU   X'002B'        NETWORK QUALIFIED NAME REQUIRED
USF6INEL EQU   X'002C'        INVALID EXPEDITED DATA LENGTH
*                             SPECIFIED
USF6INSC EQU   X'002D'        INVALID SENSE CODE SPECIFIED
USF6VANV EQU   X'002E'        VECTOR AREA NOT VALID
USF6VALI EQU   X'002F'        VECTOR AREA LENGTH INSUFFICIENT
USF6STNV EQU   X'0030'        STORAGE TYPE NOT VALID
USF6VALS EQU   X'0031'        SENDRCV SPECIFIED WITHOUT
*                             OPTCD=BUFFLST|XBUFLST
USF6UNXV EQU   X'0032'        UNEXPECTED VECTOR PROVIDED ON
*                             APPCCMD
USF6VNPV EQU   X'0033'        A REQUIRED VECTOR WAS NOT PROVIDED
*                             OR SPECIFIED INCORRECTLY
USF6LNSP EQU   X'0034'        PASSWORD SUBSTITUTION VALUE SET IN
*                             ERROR
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'005C' (USF6UECR).
***
USF6FNGR EQU   X'0000'        FOLLOWING NEGATIVE RESPONSE
USF6WNGR EQU   X'0001'        WITHOUT NEGATIVE RESPONSE
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'009C' (USF6RSTF).
***
USF6SLSR EQU   X'0001'        RESTORE ISSUED BEFORE SETLOGON START
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'00A0' (USF6RNAL).
***
USF6LNSE EQU   X'0001'        LU PAIR NOT SUPPORTING EXPEDITED
*                             DATA REQUESTS
USF6RQBL EQU   X'0002'        REQUEST BLOCKED DUE TO PENDING
*                             CONVERSATION TERMINATION
USF6RNEX EQU   X'0003'        EXECUTION OF REQUEST TERMINATED
USF6VNVF EQU   X'0004'        CONTROL/QUALIFY VALUE INVALID ON
*                             FULL-DUPLEX CONVERSATION
USF6EXRO EQU   X'0005'        EXPEDITED DATA RESPONSE OUTSTANDING
USF6NAUT EQU   X'0006'        PROGRAM NOT AUTHORIZED FOR REQUESTED
*                             FUNCTION
*              X'0007'        RESERVED
USF6ENEL EQU   X'0008'        NAMED RESOURCE NOT ELIGIBLE FOR
*                             FOR REQUESTED ALTERATION
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'00A8' (USF6ENVE).
***
USF6OSLV EQU   X'0000'        OPERATING SYSTEM LEVEL DOES NOT SUPPORT
*                             REQUESTED FUNCTION
USF6XMES EQU   X'0001'        SUSPEND FAILURE
USF6XMER EQU   X'0002'        RESUME FAILURE
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'00AC' (USF6ERIN).
***
USF6EIAP EQU   X'0001'        DEALLOCATE ABEND PROGRAM
USF6EIAS EQU   X'0002'        DEALLOCATE ABEND SERVICE
USF6EIAT EQU   X'0003'        DEALLOCATE ABEND TIMER
```

```
USF6EIAE EQU   X'0004'        ALLOCATION ERROR
USF6EIUN EQU   X'0005'        UNKNOWN TERMINATION TYPE RECEIVED
USF6EIRR EQU   X'0006'        RESOURCE FAILURE, RETRY
USF6EIRN EQU   X'0007'        RESOURCE FAILURE, NO RETRY
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'00B0' (USF6NRER).
***
USF6NRRE EQU   X'0001'        LUNAME FOUND IN A VARIANT_NAME ENTRY
USF6NRRD EQU   X'0002'        NAME RETURNED DIFFERS FROM
*                             ASSOCIATED NAME
USF6NRRA EQU   X'0003'        NAME RETURNED FOUND IN A
*                             VARIANT NAME ENTRY
USF6NRAP EQU   X'0004'        NAME RETURNED FOUND IN A
*                             SUPPLIED NAME ENTRY
USF6NRNM EQU   X'0005'        NETWORK NAME MISMATCH
USF6NRAV EQU   X'0006'        LUNAME FOUND IN AN UNUSABLE_NAME
*                             ENTRY
USF6NRIV EQU   X'0007'        NAME RETURNED FOUND IN AN
*                             UNUSABLE_NAME ENTRY
USF6NRDN EQU   X'0008'        LUNAME FOUND IN A DISASSOCIATED_NAME
*                             ENTRY
*
***
*** THE FOLLOWING ARE SECONDARY RETURN CODES WHEN THE
*** PRIMARY RETURN CODE IS SET TO X'00B4' (USF6CSME).
***
USF6NSPC EQU   X'0001'        NOT SPECIFIED
USF6IBTK EQU   X'0002'        INVALID BUFFER TOKEN SPECIFIED
USF6IIID EQU   X'0003'        INVALID INSTANCE ID SPECIFIED
*
**********************************************************************
**********************************************************************
```

APPCCMD Vector Lists (ISTAPCVL)

```
  LOC    SOURCE STATEMENT
       ***     MAPPING FOR VECTORLIST HEADER (LENGTH FIELD)                **
000000 ISTAPCVA DSECT                      VECTOR LIST
       *
000000 APCVALEN DS    HL2                  LENGTH OF VECTOR LIST
       *                                   (INCLUDES LENGTH FIELD & VECTORS)
000002 APCVADTA DS    0X                   VECTORS
       *
       **********************************************************************
       ***  GENERALIZED MAPPING FOR EXAMINING OR BUILDING COMMON FIELDS IN  **
       ***  ALL APPCCMD VECTORS IN THE VECTOR LISTS POINTED TO BY RPL6VAIA  **
       ***  AND RPL6VAIA                                                    **
       **********************************************************************
000000 ISTAPCVT DSECT                      VECTOR TEMPLATE
000000 APCVTLEN DS    HL2                  VECTOR LENGTH
000002 APCVTKEY DS    X                    VECTOR KEY
000003 APCVTDTA DS    0X                   VECTOR DATA
       *
       *
       **********************************************************************
       ***                                                                 **
       *** VECTORS PASSED FROM VTAM TO APPLICATION AT APPCCMD COMPLETION    **
       ***                                                                 **
       *** Note: Highorder bit in vector key is off for all vectors sent    **
       ***       from VTAM to application.                                 **
       ***                                                                 **
       **********************************************************************
       *
       **********************************************************************
       *** ISTAPC10 - maps the VTAM-to-APPL Required INFORMATION vector.    **
```

```
       ***          - Returned on all APPCCMD macros if a vector area is    **
       ***            provided.                                             **
       ***          - Indicates whether VTAM was able to return vector      **
       ***            information successfully and length needed.           **
       ***          - NOTE: Application-provided vector area must be large  **
       ***            enough to accept at least this vector.                **
       ************************************************************************
000000 ISTAPC10 DSECT                      INFORMATION VECTOR
       *
000000 APC10LEN DS    HL2                  VECTOR LENGTH
000002 APC10KEY DS    X                    VECTOR KEY
       APC10KYC EQU   X'10'                VECTOR KEY X'10'
000003 APC10DTA DS    0X                   VECTOR DATA FIELDS
000003 APC10FLG DS    X                    FLAGS
       APC10IVL EQU   X'80'                INSUFFICIENT VECTOR AREA LENGTH
000004          DS    X                    RESERVED
000005          DS    X                    RESERVED
000006 APC10VLN DS    HL2                  VECTOR AREA LENGTH NEEDED
       *
       *
       ************************************************************************
       *** ISTAPC12 - Maps the Partner's DCE Capability vector.             **
       ***          - Returned on these APPCCMD completions if DCE is       **
       ***            active:                                               **
       ***               APPCCMD CONTROL=PREALLOC                           **
       ***               APPCCMD CONTROL=RCVFMH5                            **
       ***               APPCCMD CONTROL=OPRCNTL QUALIFY=CNOS               **
       ***               APPCCMD CONTROL=OPRCNTL QUALIFY=DISPLAY            **
       ***          - Also returned on ATTN(CNOS) if DCE is active.         **
       ***          - Contains the Security Mechanisms Data subfield        **
       ***            exchanged during BIND processing if DCE is active.    **
       ************************************************************************
000000 ISTAPC12 DSECT                      PARTNER'S DCE CAPABILITY VECTOR
       *                                   MAPPING
000000 APC12LEN DS    HL2                  LENGTH OF VECTOR (INCLUDING
       *                                   LENGTH OF THIS FIELD)
000002 APC12KEY DS    X                    VECTOR KEY
       APC12KYC EQU   X'12'                VECTOR KEY X'12'
000003 APC12DTA DS    0X                   SECURITY MECHANISMS DATA
       APC12DCE EQU   X'01'                DCE AUTHENTICATION
       APC12KRY EQU   X'02'                KRYPTOKNIGHT
       APC12KER EQU   X'03'                KERBEROS V5
       APC12DCP EQU   X'04'                DCE PERFORMANCE MECHANISM
       *
       *
       ************************************************************************
       *** ISTAPC13 - maps the LOCAL NONCE vector.                          **
       ***          - Returned for these APPCCMD completions if             **
       ***            password substitution is supported on session:        **
       ***               APPCCMD CONTROL=PREALLOC                           **
       ***               APPCCMD CONTROL=RCVFMH5                            **
       ***          - Contains random data used for password                **
       ***            substitution.                                         **
       ************************************************************************
000000 ISTAPC13 DSECT                      MAPPING FOR LOCAL NONCE VECTOR
000000 APC13LEN DS    HL2                  LENGTH OF VECTOR
000002 APC13KEY DS    X                    VECTOR KEY
       APC13KYC EQU   X'13'                KEY IS X'13'
000003 APC13DTA DS    0X                   NONCE DATA
000003          DS    XL1                  RESERVED
000004 APC13NOF DS    CL8                  NONCE FIELD
       *
       *
       ************************************************************************
       *** ISTAPC14 - maps the PARTNER'S NONCE vector.                      **
       ***          - Returned for these APPCCMD completions if             **
       ***            password substitution is supported on session:        **
```

```
       ***              APPCCMD CONTROL=PREALLOC                          **
       ***              APPCCMD CONTROL=RCVFMH5                           **
       ***          - Contains random data used for password              **
       ***            substitution.                                       **
       ************************************************************************
000000 ISTAPC14 DSECT                     MAPPING FOR PARTNER NONCE
000000 APC14LEN DS    HL2                 LENGTH OF VECTOR
000002 APC14KEY DS    X                   VECTOR KEY
       APC14KYC EQU   X'14'               KEY IS X'14'
000003 APC14DTA DS    0X                  NONCE DATA
000003          DS    XL1                 RESERVED
000004 APC14NOF DS    CL8                 NONCE FIELD
       *
       *
       ************************************************************************
       *** ISTAPC15 - maps the SEND FMH_5 SEQUENCE NUMBER vector.          **
       ***          - Returned for these APPCCMD completions if            **
       ***            password substitution is supported on session:       **
       ***              APPCCMD CONTROL=PREALLOC                          **
       ***          - Contains the number of FMH_5s which have flowed on   **
       ***            this session from the partner LU.                   **
       ************************************************************************
000000 ISTAPC15 DSECT                     MAPPING FOR SEND FMH_5
       *                                    SEQUENCE NUMBER VECTOR
000000 APC15LEN DS    HL2                 LENGTH OF VECTOR
000002 APC15KEY DS    X                   VECTOR KEY
       APC15KYC EQU   X'15'               KEY IS X'15'
000003          DS    XL1                 RESERVED
000004 APC15SNF DS    0X                  SEQUENCE NUMBER FIELD
000004 APC15SNH DS    XL4                 SEQUENCE NUMBER FIELD -
       *                                    HIGH-ORDER BITS
000008 APC15SNL DS    XL4                 SEQUENCE NUMBER FIELD -
       *                                    LOW-ORDER BITS
       *
       *
       ************************************************************************
       *** ISTAPC16 - maps the RECEIVE FMH_5 SEQUENCE NUMBER vector.       **
       ***          - Returned for these APPCCMD completions if            **
       ***            password substitution is supported on session:       **
       ***              APPCCMD CONTROL=RCVFMH5                           **
       ***          - Contains the number of FMH_5s which have flowed on   **
       ***            this session from the partner LU.                   **
       ************************************************************************
000000 ISTAPC16 DSECT                     MAPPING FOR RECEIVE FMH_5
       *                                    SEQUENCE NUMBER VECTOR
000000 APC16LEN DS    HL2                 LENGTH OF VECTOR
000002 APC16KEY DS    X                   VECTOR KEY
       APC16KYC EQU   X'16'               KEY IS X'16'
000003          DS    XL1                 RESERVED
000004 APC16SNF DS    0X                  SEQUENCE NUMBER FIELD
000004 APC16SNH DS    XL4                 SEQUENCE NUMBER FIELD -
       *                                    HIGH-ORDER BITS
000008 APC16SNL DS    XL4                 SEQUENCE NUMBER FIELD -
       *                                    LOW-ORDER BITS
       *
       ************************************************************************
       *** ISTAPC17 - maps the PCID vector.                                **
       ***          - Returned for these APPCCMD completions:              **
       ***              APPCCMD CONTROL=ALLOC                             **
       ***              APPCCMD CONTROL=PREALLOC                          **
       ***              APPCCMD CONTROL=RCVFMH5                           **
       ***          - Contains the PCID for the session being used by the  **
       ***            conversation.                                       **
       ************************************************************************
000000 ISTAPC17 DSECT                     MAPPING FOR PCID VECTOR
000000 APC17LEN DS    HL2                 LENGTH OF VECTOR
000002 APC17KEY DS    X                   VECTOR KEY
```

```
       APC17KYC EQU   X'17'                KEY IS X'17'
000003 APC17DTA DS    0X                   VECTOR DATA FIELDS
000003 APC17PCF DS    CL8                  SESSION PCID FIELD
       *
       *
       ***********************************************************************
       *** ISTAPC18 - maps the NAME CHANGE vector.                          **
       ***          - Returned for these APPCCMD completions and exits:     **
       ***               APPCCMD CONTROL=ALLOC                              **
       ***               APPCCMD CONTROL=OPRCNTL,QUALIFY=CNOS               **
       ***               APPCCMD CONTROL=PREALLOC                           **
       ***               ATTN(CNOS) exit                                    **
       ***            ...when a RCVD_NAME LU entry has been changed to      **
       ***            a VARIANT_NAME LU entry in the LU-Mode Table.         **
       ***********************************************************************
000000 ISTAPC18 DSECT                      MAPPING FOR NAME CHANGE VECTOR
000000 APC18LEN DS    HL2                  LENGTH OF VECTOR
000002 APC18KEY DS    X                    VECTOR KEY
       APC18KYC EQU   X'18'                KEY IS X'18'
000003 APC18DTA DS    0X                   VECTOR DATA FIELDS
000003 APC18NET DS    CL8                  NETWORK IDENTIFIER OF THE LU
00000B APC18RCV DS    CL8                  LUNAME IN RCVD_NAME LU ENTRY
000013 APC18SUP DS    CL8                  LUNAME IN SUPPLIED_NAME ENTRY
       *
       ***********************************************************************
       *** ISTAPC19 - maps the Session Information vector.                  **
       ***          - Returned for these APPCCMD completions:               **
       ***               APPCCMD CONTROL=ALLOC                              **
       ***               APPCCMD CONTROL=PREALLOC                           **
       ***               APPCCMD CONTROL=RCVFMH5                            **
       ***            ...to provide session characteristics information     **
       ***            for the conversation.                                 **
       ***********************************************************************
000000 ISTAPC19 DSECT                      MAPPING FOR SESSION INFORMATION
       *                                     VECTOR
000000 APC19LEN DS    HL2                  LENGTH OF VECTOR
000002 APC19KEY DS    X                    VECTOR KEY
       APC19KYC EQU   X'19'                KEY IS X'19'
000003 APC19DTA DS    0X                   VECTOR DATA FIELDS
000003 APC19CSU DS    X                    COMMUNICATION STORAGE USAGE
       *                                   INDICATORS
       APC19NOF EQU   X'80'                NOT AN HPDT-ENABLED SESSION.
       *                                     CSM STORAGE USERS, DUE TO
       *                                     PERFORMANCE CONSTRAINTS, SHOULD
       *                                     EITHER USE CSM PAGEABLE DATA
       *                                     SPACE OR NON_CSM STORAGE
       APC19SMB EQU   X'40'                SMALLER BUFFERS RECOMMENDED FOR
       *                                     CSM STORAGE USERS BECAUSE OF
       *                                     RU SIZE LIMITATIONS.
       APC19PGP EQU   X'20'                PAGEABLE BUFFERS RECOMMENDED.  HPDT
       *                                     ENABLED FOR THIS SESSION.
       *                                     NO ADDITIONAL PERFORMANCE CAN BE
       *                                     GAINED USING FIXED BUFFERS.
       APC19FXP EQU   X'10'                FIXED BUFFERS RECOMMENDED.  HPDT
       *                                     ENABLED FOR THIS SESSION.
       *                                     ADDITIONAL PERFORMANCE CAN BE
       *                                     GAINED USING FIXED BUFFERS.
000004          DS    XL2                  RESERVED
000006 APC19RUO DS    FL4                  MAXIMUM RU SIZE OUTBOUND
00000A APC19RUI DS    FL4                  MAXIMUM RU SIZE INBOUND
       *
       *
       ***********************************************************************
       *** ISTAPC1A - maps the Partner Application Capabilities vector      **
       ***          - Returned for these APPCCMD completions:               **
       ***               APPCCMD CONTROL=ALLOC                              **
       ***               APPCCMD CONTROL=OPRCNTL, QUALIFY=CNOS              **
```

```
       ***              APPCCMD CONTROL=OPRCNTL, QUALIFY=DISPLAY          **
       ***              APPCCMD CONTROL=PREALLOC                          **
       ***              APPCCMD CONTROL=RCVFMH5                           **
       ***          - Returned for this exit:                             **
       ***              ATTN(CNOS)                                        **
       ***                                                                **
       ***            ...to provide partner capabilities information      **
       ***            for the conversation.                               **
       ***********************************************************************
000000 ISTAPC1A DSECT                      MAPPING FOR PARTNER APPLICATION
       *                                     CAPABILITIES VECTOR
000000 APC1ALEN DS    HL2                  LENGTH OF VECTOR
000002 APC1AKEY DS    X                    VECTOR KEY
       APC1AKYC EQU   X'1A'                KEY IS X'1A'
000003 APC1ADTA DS    0X                   VECTOR DATA FIELDS
000003 APC1AFL1 DS    X                    PARTNER APPLICATION CAPABILITY
       *                                     INDICATORS
       APC1APAR EQU   X'C0'                NEGOTIATED PARALLEL SESSION
       *                                     CAPABILITY
       APC1ASSC EQU   X'00'                SINGLE SESSION CAPABLE
       APC1ASSP EQU   X'40'                PENDING SINGLE STATE
       APC1APSP EQU   X'80'                PENDING PARALLEL STATE
       APC1APSC EQU   X'C0'                PARALLEL SESSION CAPABLE
       APC1APWS EQU   X'30'                NEGOTIATED LEVEL OF
       *                                     PASSWORD SUBSTITUTION
       APC1APSS EQU   X'20'                PASSWORD SUBSTITUTION
       *                                     SUPPORTED
       APC1APSN EQU   X'10'                PASSWORD SUBSTITUTION
       *                                     NOT SUPPORTED
       APC1APSU EQU   X'00'                PASSWORD SUBSTITUTION
       *                                     LEVEL NOT SET
       APC1AESS EQU   X'0C'                PARTNER SUPPORT FOR
       *                                     EXTENDED SECURITY SENSE
       *                                     CODES
       APC1ASSS EQU   X'08'                EXTENDED SECURITY SENSE CODES
       *                                      SUPPORTED
       APC1ASSN EQU   X'04'                EXTENDED SECURITY SENSE CODES
       *                                      NOT SUPPORTED
       APC1ASSU EQU   X'00'                EXTENDED SECURITY SENSE CODE
       *                                      LEVEL NOT SET
       APC1AFDX EQU   X'03'                NEGOTIATED FDX/EXPD
       *                                     CAPABILITY
       APC1AFXS EQU   X'02'                FDX OR HDX CONVERSATIONS AND
       *                                     EXPEDITED DATA ALLOWED
       APC1AFXN EQU   X'01'                HDX CONVERSATIONS ONLY
       APC1AFXU EQU   X'00'                CAPABILITY IS UNKNOWN
000004 APC1AFL2 DS    X                    PARTNER APPLICATION CAPABILITY
       *                                     INDICATORS
       APC1ACON EQU   X'C0'                NEGOTIATED LEVEL OF
       *                                     SYNCHRONIZATION
       APC1ACNS EQU   X'80'                CONFIRM, SYNC POINT AND
       *                                     BACKOUT SUPPORTED
       APC1ACNN EQU   X'40'                CONFIRM SUPPORTED
       APC1ACNU EQU   X'00'                SYNCHRONIZATION LEVEL NOT
       *                                     SET
       APC1ASEC EQU   X'20'                PARTNER ACCEPTS SECURITY
       *                                     SUBFIELDS ON FMH
       APC1AALV EQU   X'10'                PARTNER ACCEPTS REQUEST FOR
       *                                     ALREADY VERIFIED
       APC1APRV EQU   X'08'                PARTNER ACCEPTS REQUEST FOR
       *                                     PERSISTENT VERIFICATION
       *        EQU   X'07'                RESERVED
000005 APC1AFL3 DS    X                    PARTNER CHARACTERISTICS
       APC1ALOC EQU   X'E0'                PARTNER LOCALITY STATUS
       APC1AUNL EQU   X'00'                LOCALITY OF PARTNER UNKNOWN
       APC1ARMT EQU   X'80'                PARTNER NOT ON SAME HOST
       APC1ALCL EQU   X'40'                PARTNER IS ON SAME HOST SYSTEM
```

```
       APC1ALUO EQU   X'20'                PARTNER LU SAME AS APPLICATION LU
       *                                    (LU=OWN)
       *
       *
       ************************************************************************
       ***                                                                  **
       *** VECTORS PASSED FROM APPLICATION TO VTAM AT APPCCMD ISSUANCE      **
       ***                                                                  **
       *** Note: Highorder bit in vector key is on for all vectors sent     **
       ***       from application to VTAM.                                  **
       ***                                                                  **
       ************************************************************************
       *
       ************************************************************************
       *** ISTAPC82 - maps the XBUFLST RECEIVE vector.                      **
       ***          - This vector is passed to VTAM on an APPCCMD           **
       ***            CONTROL=RECEIVE when OPTCD specifies XBUFLST.         **
       ************************************************************************
000000 ISTAPC82 DSECT                      MAPPING FOR XBUFLST RECEIVE VECTOR
000000 APC82LEN DS    HL2                  LENGTH OF VECTOR
000002 APC82KEY DS    X                    VECTOR KEY
       APC82KYC EQU   X'82'                KEY IS X'82'
000003 APC82DTA DS    0X                   VECTOR DATA FIELDS
000003 APC82SFL DS    X                    STORAGE TYPE FLAG BYTE:
       *                                    ONE OR MORE OF THE FOLLOWING
       *                                    IS REQUIRED:
       APC82ECS EQU   X'80'                ECSA STORAGE REQUESTED
       APC82CDS EQU   X'40'                DATA SPACE STORAGE REQUESTED
000004 APC82XBL DS    FL4                  BUFFER LENGTH (REQUIRED WHEN
       *                                    IN FILL=BUFF MODE) OR ZEROS
       *
000008 APC82MXD DS    FL4                  MAXIMUM DATA TO BE RECEIVED
       *                                    (OPTIONAL) OR ZEROS
       *
00000C APC82TSK DS    AL4                  TASK TCB ADDRESS FOR CSM
       *                                    STORAGE ASSOCIATION
       *                                    (OPTIONAL) OR ZEROS
       *
```

Application-ACB Vector List (ISTVACBV)

```
  LOC    SOURCE STATEMENT
       ************************************************************************
       ***                                                                  **
       ***       DATA FIELDS PASSED FROM THE APPLICATION TO VTAM.           **
       ***                                                                  **
       ***                                                                  **
       *** Addressability: ACBAPID, ACBPASSW.                               **
       ***                                                                  **
       ************************************************************************
000000 ISTVACAP  DSECT                     APPLID MAPPING
       *
000000 VACAPLEN DS    X                    MAP LENGTH
000001 VACAPDTA DS    0X                   MAP DATA
       *
000000 ISTVACPW DSECT                      PASSWORD MAPPING
       *
000000 VACPWLEN DS    X                    MAP LENGTH
000001 VACPWDTA DS    0X                   MAP DATA
       *
       ************************************************************************
       ***                                                                  **
       ***          VECTORS PASSED FROM THE APPLICATION TO VTAM.            **
       ***                                                                  **
       ***                                                                  **
       *** Addressability: ACBAVPTR.                                        **
       ***                                                                  **
```

```
        *** Note: Highorder bit in vector key is on for all vectors sent     **
        ***       from application to VTAM.                                  **
        ***                                                                  **
        ************************************************************************
        *
        ***     MAPPING FOR VECTORLIST HEADER (LENGTH FIELD)                 **
000000 ISTVACAV DSECT                      APPLICATION VECTORLIST
        *                                  POINTED TO BY ACBAPVTR
        *                                  WHEN PARMS=(APPLVCTR=address)
000000 VACAVLEN DS    HL2                  TOTAL LENGTH OF APPL VECTORS
000002 VACAVDTA DS    0X                   VECTOR DATA
        *
        ************************************************************************
        ***  GENERALIZED MAPPING FOR EXAMINING OR BUILDING COMMON FIELDS IN  **
        ***  ALL ACB VECTORS IN THE VECTOR LIST POINTED TO BY ACBAVPTR       **
        ************************************************************************
000000 ISTVACVT DSECT                      VECTOR TEMPLATE
000000 VACVTLEN DS    HL2                  VECTOR LENGTH
000002 VACVTKEY DS    X                    VECTOR KEY
000003 VACVTDAT DS    0X                   VECTOR DATA
        *
        *
        ************************************************************************
        *** ISTVAC81 - Application Capabilities vector                        **
        ***          - Passed to VTAM by the application at OPEN invocation  **
        ***            for the ACB.                                          **
        ***          - Bit indicators which enable/disable application use   **
        ***            of certain VTAM functions.                            **
        ************************************************************************
000000 ISTVAC81 DSECT                      APPLICATION CAPABILITIES VECTOR
000000 VAC81LEN DS    HL2                  VECTOR LENGTH
000002 VAC81KEY DS    X                    VECTOR KEY
        VAC81KYC EQU   X'81'               KEY IS X'81'
000003 VAC81CAP DS    0XL4                 APPLICATION CAPABILITIES DATA
        VAC81MLE EQU   X'80'               APPLICATION SUPPORTS HAVING ITS
        *                                  LOGON EXIT DRIVEN MULTIPLE TIMES
        *                                  PER SESSION REQUEST.  APPLICATIONS
        *                                  WITH LOGON EXITS MUST SET THIS
        *                                  INDICATOR TO BENEFIT FROM
        *                                  VERIFICATION REDUCTION
        VAC81FPR EQU   X'40'               APPLICATION INDICATES THAT IT WILL
        *                                  USE HPDT INTERFACE PROVIDED
        *                                  VIA THE OPTCD=XBUFLST FIELD ON THE
        *                                  APPCCMD RECEIVE MACROINSTRUCTION
        *
        VAC81PWS EQU   X'20'               APPLICATION INDICATES THAT IT
        *                                  IS PASSWORD SUBSTITUTION
        *                                  CAPABLE
        VAC81ESS EQU   X'10'               APPLICATION INDICATES THAT IT
        *                                  IS CAPABLE OF EXTENDED
        *                                  SECURITY SENSE CODES
        VAC81FPS EQU   X'08'               APPLICATION INDICATES THAT IT
        *                                  WILL USE HPDT INTERFACE
        *                                  PROVIDED BY THE OPTCD=XBUFLST
        *                                  FIELD ON AN APPCCMD
        *                                  MACROINSTRUCTION THAT SENDS
        *                                  DATA
        *
        *
        ************************************************************************
        *** ISTVAC82 - Local Application's DCE Capability Vector              **
        ***          - Passed to VTAM by the application at OPEN invocation  **
        ***            for the ACB.                                          **
        ***          - Contains the Security Mechanisms data for the Local   **
        ***            LU.                                                   **
        ************************************************************************
000000 ISTVAC82 DSECT                      LOCAL APPLICATION'S DCE
```

```
       *                                CAPABILITY VECTOR MAPPING
000000 VAC82LEN DS    HL2               LENGTH OF VECTOR (INCLUDING
       *                                LENGTH OF THIS FIELD).
000002 VAC82KEY DS    X                 VECTOR KEY
       VAC82KYC EQU   X'82'             VECTOR KEY X'82'
000003 VAC82DTA DS    0X                ISTVAC82 DATA
```

Access-Method-Support Vector List (ISTAMSVL)

```
  LOC    SOURCE STATEMENT
000000 ISTAMSVL DSECT                   MAPPING FOR RESOURCE INFORMATION
       *                                VECTOR LIST POINTED TO BY ACVAMSVL
000000 AMSLLEN  DS    HL2               TOTAL LENGTH OF VECTORS
000002 AMSLDATA DS    0X                VECTOR DATA
       *
       ************************************************************************
       ***  GENERALIZED MAPPING FOR EXAMINING COMMON FIELDS IN ALL ACB      **
       ***  VECTORS IN THE VECTOR LIST POINTED TO BY ACBAMSVL               **
       ************************************************************************
000000 ISTAMSVT DSECT                   VECTOR FIELDS
000000 AMSVTLEN DS    X                 VECTOR LENGTH
000001 AMSVTKEY DS    X                 VECTOR KEY
000002 AMSVTDAT DS    0X                VECTOR DATA
       *
       *
       ************************************************************************
       *** ISTAMS01 - maps the RELEASE LEVEL vector.                        **
       ***          - Contains identification codes for the access method   **
       ***            product and its version, release, and modification    **
       ***            level.                                                **
       ************************************************************************
000000 ISTAMS01 DSECT                   RELEASE LEVEL VECTOR
000000 AMS01LEN DS    X                 VECTOR LENGTH
000001 AMS01KEY DS    X                 VECTOR KEY
       AMS01KYC EQU   X'01'             KEY IS X'01'
000002 AMS01DTA DS    0CL4              VECTOR DATA
000002 AMS01PRD DS    CL1               PRODUCT CODE
       AMS01VTM EQU   C'0'              VTAM PRODUCT CODE
000003 AMS01VER DS    CL1               VERSION CODE
000004 AMS01REL DS    CL1               RELEASE CODE
000005 AMS01MDF DS    CL1               MODIFICATION CODE
       *
       ************************************************************************
       *** ISTAMS04 - maps the COMPONENT IDENTIFICATION vector.             **
       ***          - This vector may be repeated.                          **
       ***          - Each component identification vector contains product **
       ***            identification information about a major component or **
       ***            feature of the VTAM licensed program.  When multiple  **
       ***            component identification vectors are present, the     **
       ***            first one designates the base VTAM product and later  **
       ***            vectors are features or other major VTAM components.  **
       ***          - The vector data is in the form: C'xxxx-xxxxx-xxx'.    **
       ************************************************************************
000000 ISTAMS04 DSECT                   COMPONENT IDENTIFICATION VECTOR
000000 AMS04LEN DS    X                 VECTOR LENGTH
000001 AMS04KEY DS    X                 VECTOR KEY
       AMS04KYC EQU   X'04'             KEY IS X'04'
000002 AMS04DTA DS    CL14              VECTOR DATA
       *
       ************************************************************************
       *** ISTAMS05 - maps the FUNCTION LIST vector.                        **
       ***          - The vector data is a variable-length bit string, in   **
       ***            which each bit corresponds to a particular VTAM       **
       ***            function.  If a bit is on, the corresponding function **
       ***            is present in the executing release of VTAM.  If a    **
       ***            bit is off, the function is not available.  If the    **
       ***            vector is not present or if the bit string is shorter **
```

```
        ***            than expected, you may assume that the missing bits   **
        ***            are zero and their corresponding functions are not    **
        ***            available.                                            **
        ***          - These indicator bits correspond to the compile-time   **
        ***            global indicator bits in the ISTGLBAL macro.          **
        ************************************************************************
000000 ISTAMS05 DSECT                      FUNCTION LIST VECTOR
000000 AMS05LEN DS    X                    VECTOR LENGTH
000001 AMS05KEY DS    X                    VECTOR KEY
       AMS05KYC EQU   X'05'                KEY IS X'05'
000002 AMS05DTA DS    0X                   VECTOR DATA
000002 AMS05DT0 DS    X                    BYTE 0 OF INDICATORS
       AMS05B00 EQU   X'80'                NIB ENCR AND RPL CRYPT
       *                                   (CRYPTOGRAPHY)
       AMS05B01 EQU   X'40'                ACB PARMS=NIB (COMMUNICATION
       *                                   NETWORK MANAGEMENT INTERFACE)
       AMS05B02 EQU   X'20'                MULTIPLE-ADDRESS-SPACE
       *                                   APPLICATIONS PROGRAMS
       AMS05B03 EQU   X'10'                AUTHORIZED PATH FOR
       *                                   COMMUNICATIONS MACROS
       AMS05B04 EQU   X'08'                AUTHORIZED PATH FOR ALL
       *                                   RPL-BASED MACROS
       AMS05B05 EQU   X'04'                SRBEXIT (ON APPL DEFINITION
       *                                   STATEMENT)
       AMS05B06 EQU   X'02'                SONSCIP (ON APPL DEFINITION
       *                                   STATEMENT)
       AMS05B07 EQU   X'01'                VTAMFRR (ON APPL DEFINITION
       *                                   STATEMENT)
       *
000003 AMS05DT1 DS    X                    BYTE 1 OF INDICATORS
       AMS05B10 EQU   X'80'                SSCP TRACKING OF DEVICE-LU
       *                                   SESSION CAPABILITY VIA NOTIFY
       *                                   (ENABLED/DISABLED/INHIBITED)
       AMS05B11 EQU   X'40'                RPL OPTCD=LMPEO
       AMS05B12 EQU   X'20'                RPL OPTCD=BUFFLST
       AMS05B13 EQU   X'10'                RPL OPTCD=USERRH
       AMS05B14 EQU   X'08'                ACB PARMS=USERFLD
       AMS05B15 EQU   X'04'                RPL BRACKET=CEB
       AMS05B16 EQU   X'02'                APPLICATION PROGRAM ASSIGNMENT OF
       *                                   SEQUENCE NUMBERS FOR EXPEDITED DFC
       AMS05B17 EQU   X'01'                RESOURCE-IDENTIFICATION VECTOR LIST
       *
000004 AMS05DT2 DS    X                    BYTE 2 OF INDICATORS
       AMS05B20 EQU   X'80'                ACCESS-METHOD-SUPPORT VECTOR LIST
       AMS05B21 EQU   X'40'                RETURN OF SYSTEM RESPONSE BYTE AND
       *                                   EXTENDED RESPONSE BYTE FOR BSC 3270
       *                                   TERMINALS ATTACHED TO ACF/NCP
       AMS05B22 EQU   X'20'                INTRPRET
       AMS05B23 EQU   X'10'                VTAM API IS XRF CAPABLE
       AMS05B24 EQU   X'08'                SENSE ON -RSP(CINIT). CLSDST
       *                                   OPTCD=(RELEASE,SENSE)
       AMS05B25 EQU   X'04'                UNBIND SON CODE AND SENSE.
       *                                   CLSDST OPTCD=(RELEASE,SONCODE),
       *                                   TERMSESS OPTCD=(UNBIND,SONCODE)
       AMS05B26 EQU   X'02'                HOLD/RELEASE LOGON/SCIP EXIT FOR
       *                                   SESSION SETUP.
       *                                   SETLOGON OPTCD=(START|HOLD)
       AMS05B27 EQU   X'01'                CINIT - NETWORK ADDRESSES IN
       *                                   VECTOR KEY X'15'
       *
000005 AMS05DT3 DS    X                    BYTE 3 OF INDICATORS
       AMS05B30 EQU   X'80'                31-BIT API
       AMS05B31 EQU   X'40'                NOTIFICATION OF QUEUED RESPONSES
       *                                   SUPPORTED.  SEND OPTCD=(RSPQUED)
       AMS05B32 EQU   X'20'                APPC IS SUPPORTED
       AMS05B33 EQU   X'10'                ADD SUPPORT FOR USERVAR
       AMS05B34 EQU   X'08'                VCNS API SUPPORT FOR X.25
```

```
       AMS05B35 EQU   X'04'                VCNS API SUPPORT FOR TOKEN BUS,
       *                                   TOKEN RING,
       AMS05B36 EQU   X'02'                CROSS-MEMORY API IS SUPPORTED
       AMS05B37 EQU   X'01'                KEEPFRR SUPPORT (ON ACB STATEMENT)
       *
000006 AMS05DT4 DS    X                    BYTE 4 OF INDICATORS
       AMS05B40 EQU   X'80'                SRBEXIT SUPPORT (ON ACB STATEMENT)
       AMS05B41 EQU   X'40'                PERSISTENT LU-LU SESSIONS
       AMS05B42 EQU   X'20'                V.25BIS SUPPORT
       AMS05B43 EQU   X'10'                VTAM/NPM INTERFACE SUPPORT
       AMS05B44 EQU   X'08'                LU6 PLUS TRACKING SUPPORTED
       AMS05B45 EQU   X'04'                BYTE 4, BIT 5: RESERVED
       AMS05B46 EQU   X'02'                BYTE 4, BIT 6: RESERVED
       AMS05B47 EQU   X'01'                NETWORK QUALIFIED NAMES SUPPORTED
       *
000007 AMS05DT5 DS    X                    BYTE 5 OF INDICATORS
       AMS05B50 EQU   X'80'                MS TRANSPORT SUPPORTED
       AMS05B51 EQU   X'40'                PERFORMANCE MONITOR INTERFACE
       *                                   SUPPORTED
       AMS05B52 EQU   X'20'                QUEUED SESSION TERMINATION
       *                                   SUPPPORTED
       AMS05B53 EQU   X'10'                VTAM AGENT SUPPORTED
       AMS05B54 EQU   X'08'                GENERIC RESOURCES SUPPORTED
       AMS05B55 EQU   X'04'                OPTCD=KEEPSRB FOR SYNC SRB
       *                                   SUSPEND/RESUME
       AMS05B56 EQU   X'02'                APPLICATION VECTORS SUPPORTED ON
       *                                   ACB MACRO
       AMS05B57 EQU   X'01'                SETLOGON GNAMESUB SUPPORTED
       *
000008 AMS05DT6 DS    X                    BYTE 6 OF INDICATORS
       AMS05B60 EQU   X'80'                BYTE 6, BIT 0: RESERVED
       AMS05B61 EQU   X'40'                BYTE 6, BIT 1: RESERVED
       AMS05B62 EQU   X'20'                BYTE 6, BIT 2: RESERVED
       AMS05B63 EQU   X'10'                BYTE 6, BIT 3: RESERVED
       AMS05B64 EQU   X'08'                BYTE 6, BIT 4: RESERVED
       AMS05B65 EQU   X'04'                BYTE 6, BIT 5: RESERVED
       AMS05B66 EQU   X'02'                BYTE 6, BIT 6: RESERVED
       AMS05B67 EQU   X'01'                BYTE 6, BIT 7: RESERVED
       *
000009 AMS05DT7 DS    X                    BYTE 7 OF INDICATORS
       AMS05B70 EQU   X'80'                BYTE 7, BIT 0: RESERVED
       AMS05B71 EQU   X'40'                BYTE 7, BIT 1: RESERVED
       AMS05B72 EQU   X'20'                BYTE 7, BIT 2: RESERVED
       AMS05B73 EQU   X'10'                BYTE 7, BIT 3: RESERVED
       AMS05B74 EQU   X'08'                BYTE 7, BIT 4: RESERVED
       AMS05B75 EQU   X'04'                BYTE 7, BIT 5: RESERVED
       AMS05B76 EQU   X'02'                BYTE 7, BIT 6: RESERVED
       AMS05B77 EQU   X'01'                BYTE 7, BIT 7: RESERVED
       *
       ************************************************************************
       *** ISTAMS06 - maps the LU6.2 SUPPORT FUNCTION LIST vector.          **
       ***          - The vector data is a variable-length string of byte   **
       ***            indicators, each byte corresponding to a particular   **
       ***            LU6.2 function.  Each byte will have a value showing  **
       ***            that its corresponding function is either supported,  **
       ***            not supported, or supported on a "pass-through" basis.**
       ***            (Pass-through functions are those which VTAM does not **
       ***            directly provide but provides the ability for the     **
       ***            application to create the support itself.)            **
       ***            If the vector is not present or if the byte string    **
       ***            is shorter than expected, you may assume that the     **
       ***            missing bytes are zero and their corresponding        **
       ***            functions are not supported.                          **
       ***          - These indicator bytes correspond to the compile-time  **
       ***            global indicators in the ISTGAPPC macro.              **
       ************************************************************************
000000 ISTAMS06 DSECT                      LU6.2 SUPPORT FUNCTION LIST VECTOR
```

Access-Method-Support Vector List (ISTAMSVL)

```
000000 AMS06LEN DS    X                    VECTOR LENGTH
000001 AMS06KEY DS    X                    VECTOR KEY
       AMS06KYC EQU   X'06'                KEY IS X'06'
000002 AMS06DTA DS    0X                   VECTOR DATA
000002 AMS06D01 DS    X                    01. CONVERSATIONS BETWEEN TPS
       *                                       AT SAME LU
000003 AMS06D02 DS    X                    02. DELAYED SESSION
       *                                       ALLOCATION
000004 AMS06D03 DS    X                    03. IMMEDIATE SESSION
       *                                       ALLOCATION
000005 AMS06D04 DS    X                    04. SYNC POINT SERVICES
000006 AMS06D05 DS    X                    05. PROGRAM RECONNECT
000007 AMS06D06 DS    X                    06. RESERVED
000008 AMS06D07 DS    X                    07. SESSION-LEVEL LU-LU
       *                                       VERIFICATION
000009 AMS06D08 DS    X                    08. USERID VERIFICATION
00000A AMS06D09 DS    X                    09. PROGRAM SUPPLIED USERID
       *                                       AND PASSWORD
00000B AMS06D10 DS    X                    10. USERID AUTHORIZATION
00000C AMS06D11 DS    X                    11. PROFILE VERIFICATION AND
       *                                       AUTHORIZATION
00000D AMS06D12 DS    X                    12. RESERVED
00000E AMS06D13 DS    X                    13. PROFILE PASSTHROUGH
00000F AMS06D14 DS    X                    14. PROGRAM-SUPPLIED PROFILE
000010 AMS06D15 DS    X                    15. SEND PERSISTENT
       *                                       VERIFICATION
000011 AMS06D16 DS    X                    16. RECEIVE PERSISTENT
       *                                       VERIFICATION
000012 AMS06D17 DS    X                    17. PIP DATA
000013 AMS06D18 DS    X                    18. LOGGING OF DATA IN SYSTEM
       *                                       LOG
000014 AMS06D19 DS    X                    19. FLUSH LU SEND BUFFER
000015 AMS06D20 DS    X                    20. LUW IDENTIFIER
000016 AMS06D21 DS    X                    21. PREPARE TO RECEIVE
000017 AMS06D22 DS    X                    22. LONG LOCKS
000018 AMS06D23 DS    X                    23. POST ON RECEIPT WITH WAIT
000019 AMS06D24 DS    X                    24. POST ON RECEIPT WITH TEST
       *                                       FOR POSTING
00001A AMS06D25 DS    X                    25. RECEIVE-IMMEDIATE
00001B AMS06D26 DS    X                    26. TEST FOR REQUEST-TO-SEND
       *                                       RECEIVED
00001C AMS06D27 DS    X                    27. DATA MAPPING
00001D AMS06D28 DS    X                    28. FMH APPLICATION-DATA
00001E AMS06D29 DS    X                    29. GET ATTRIBUTES
00001F AMS06D30 DS    X                    30. GET CONVERSATION-TYPE
000020 AMS06D31 DS    X                    31. MAPPED CONVERSATION LU
       *                                       SERVICES COMPONENT
000021 AMS06D32 DS    X                    32. CHANGE_SESSION_LIMIT VERB
000022 AMS06D33 DS    X                    33. MIN_CONWINNERS_TARGET
       *                                       PARAMETER
000023 AMS06D34 DS    X                    34. RESPONSIBLE(TARGET)
       *                                       PARAMETER
000024 AMS06D35 DS    X                    35. DRAIN_TARGET(NO) PARAMETER
000025 AMS06D36 DS    X                    36. FORCE PARAMETER
000026 AMS06D37 DS    X                    37. ACTIVATE_SESSION VERB
000027 AMS06D38 DS    X                    38. DEACTIVATE_SESSION VERB
000028 AMS06D39 DS    X                    39. LU-PARAMETER VERBS
000029 AMS06D40 DS    X                    40. LU-LU SESSION LIMIT
00002A AMS06D41 DS    X                    41. LOCALLY-KNOWN LU NAMES
00002B AMS06D42 DS    X                    42. UNINTERPRETED LU NAMES
00002C AMS06D43 DS    X                    43. SINGLE-SESSION
       *                                       RE-INITIATION
00002D AMS06D44 DS    X                    44. ALTERNATE CODE PROCESSING
00002E AMS06D45 DS    X                    45. MAXIMUM RU SIZE BOUNDS
00002F AMS06D46 DS    X                    46. SESSION-LEVEL MANDATORY
       *                                       CRYPTOGRAPHY
000030 AMS06D47 DS    X                    47. CONTENTION WINNER
```

```
       *                                        AUTOMATIC ACTIVATION LIMIT
000031 AMS06D48 DS    X                    48. CONWINNER SESSION
       *                                        ALLOCATION
000032 AMS06D49 DS    X                    49. ENHANCED SECURITY (SAME)
000033 AMS06D50 DS    X                    50. SESSION-LEVEL SELECTIVE
       *                                        CRYPTOGRAPHY
000034 AMS06D51 DS    X                    51. CONVERSATION GROUP SUPPORT
000035 AMS06D52 DS    X                    52. ALLOCATE WHEN SESSION FREE
000036 AMS06D53 DS    X                    53. FULL-DUPLEX
000037 AMS06D54 DS    X                    54. APPCCMD VECTOR LISTS
000038 AMS06D55 DS    X                    55. QUEUED RCVFMH5
000039 AMS06D56 DS    X                    56. HIGH PERFORMANCE DATA
       *                                        TRANSFER
00003A AMS06D57 DS    X                    57. APPCCMD SENDRCV
00003B AMS06D58 DS    X                    58. INTRA-LU CONVERSATIONS
00003C AMS06D59 DS    X                    59. PASSWORD SUBSTITUTION
00003D AMS06D60 DS    X                    60. EXTENDED SECURITY SENSE
00003E AMS06D61 DS    X                    61. DCE SECURITY SERVICES
       *
```

Resource-Information Vector List (ISTRIVL)

```
  Loc    Source Statement
000000 ISTRIVL  DSECT                     MAPPING FOR RESOURCE INFORMATION
       *                                  VECTOR LIST POINTED TO BY ACBRIVL
000000 RIVLLEN  DS    HL2                 TOTAL LENGTH OF VECTORS
000002 RIVLDATA DS    0X                  VECTOR DATA
       *
       ************************************************************************
       ***  GENERALIZED MAPPING FOR EXAMINING COMMON FIELDS IN ALL ACB      **
       ***  VECTORS IN THE VECTOR LIST POINTED TO BY ACBRIVL                **
       ************************************************************************
000000 ISTRIVVT DSECT                     VECTOR TEMPLATE                @Y3A
000000 RIVVTLEN DS    X                   VECTOR LENGTH                  @Y3A
000001 RIVVTKEY DS    X                   VECTOR KEY                     @Y3A
000002 RIVVTDAT DS    0X                  VECTOR DATA                    @Y3A
       *
       *
       ************************************************************************
       *** ISTRIV02 - maps the application's network name vector.           **
       ***          - The name is specified by the name field of the        **
       ***            application definition statement.                     **
       ***          - This is obtained from the NAME ON APPL STATEMENT.     **
       ************************************************************************
000000 ISTRIV02 DSECT                     APPLICATION NETWORK NAME VECTOR
       *                                  (FROM NAME ON APPL STATEMENT)
000000 RIV02LEN DS    X                   VECTOR LENGTH
000001 RIV02KEY DS    X                   VECTOR KEY
       RIV02KYC EQU   X'02'               KEY IS X'02'
000002 RIV02DTA DS    CL8                 VECTOR DATA
       *
       ************************************************************************
       *** ISTRIV03 - maps the application's ACB name vector.               **
       ***          - This is supplied by the APPLID operand on the ACB     **
       ***            statement or can be supplied by the operating         **
       ***            system.  During OPEN ACB, VTAM will search for the    **
       ***            application's characteristics by matching the ACB     **
       ***            APPLID value to an RDTE with the application's        **
       ***            ACBNAME.  If ACBNAME was not coded for the            **
       ***            application, VTAM will search for a match with an     **
       ***            RDTE containing the application's network name.       **
       ***          - This is obtained from the APPLID on ACB MACRO.        **
       ************************************************************************
000000 ISTRIV03 DSECT                     APPLICATION ACB NAME VECTOR
       *                                  (FROM APPLID ON ACB MACRO)
000000 RIV03LEN DS    X                   VECTOR LENGTH
000001 RIV03KEY DS    X                   VECTOR KEY
```

```
       RIV03KYC EQU   X'03'                KEY IS X'03'
000002 RIV03DTA DS    CL8                  VECTOR DATA
       *
       ***********************************************************************
       *** ISTRIV06 - maps the network name in which the host resides.      **
       ***          - This is obtained from the NETID START OPTION.         **
       ***            If NETID start option is not specified, this value    **
       ***            will be blanks.                                       **
       ***********************************************************************
000000 ISTRIV06 DSECT                      NETWORK NAME VECTOR
       *                                   (FROM NETID START OPTION)
000000 RIV06LEN DS    X                    VECTOR LENGTH
000001 RIV06KEY DS    X                    VECTOR KEY
       RIV06KYC EQU   X'06'                KEY IS X'06'
000002 RIV06DTA DS    CL8                  VECTOR DATA
       *
       ***********************************************************************
       *** ISTRIV07 - maps the SSCP Name vector.                            **
       ***          - This is obtained from the SSCPNAME START OPTION       **
       ***********************************************************************
000000 ISTRIV07 DSECT                      SSCP NAME VECTOR
       *                                   (FROM SSCPNAME START OPTON)
000000 RIV07LEN DS    X                    VECTOR LENGTH
000001 RIV07KEY DS    X                    VECTOR KEY
       RIV07KYC EQU   X'07'                KEY IS X'07'
000002 RIV07DTA DS    CL8                  VECTOR DATA
       *                                   (DEFAULT IS 'VTAM')
       *
       ***********************************************************************
       *** ISTRIV08 - maps the Host Subarea PU Network Name vector.         **
       ***          - This is obtained from the HOSTPU START OPTION         **
       ***            If HOSTPU start option is not specified, the name     **
       ***            will default to 'ISTPUS  '.                           **
       ***********************************************************************
000000 ISTRIV08 DSECT                      HOST SUBAREA PU NETWORK NAME VECTOR
       *                                   (FROM HOSTPU START OPTION)
000000 RIV08LEN DS    X                    VECTOR LENGTH
000001 RIV08KEY DS    X                    VECTOR KEY
       RIV08KYC EQU   X'08'                KEY IS X'08'
000002 RIV08DTA DS    CL8                  VECTOR DATA
       *                                   (DEFAULT IS 'ISTPUS')
       *
       ***********************************************************************
       *** ISTRIV09 - maps the Host Subarea PU network address vector.      **
       ***          - It contains the network address of the host           **
       ***            subarea PU.                                           **
       ***********************************************************************
000000 ISTRIV09 DSECT                      HOST SUBAREA PU NETWORK ADDRESS
       *
000000 RIV09LEN DS    X                    VECTOR LENGTH
000001 RIV09KEY DS    X                    VECTOR KEY
       RIV09KYC EQU   X'09'                KEY IS X'09'
000002 RIV09DTA DS    XL6                  VECTOR DATA
       *
       ***********************************************************************
       *** ISTRIV0A - maps the maximum subarea vector.                      **
       ***          - Contains the maximum subarea number that is valid     **
       ***            for the host's domain.                                **
       ***          - This is obtained from the MAXSUBA START OPTION        **
       ***********************************************************************
000000 ISTRIV0A DSECT                      MAXIMUM SUBAREA NUMBER VECTOR
       *                                   (FROM MAXSUBA START OPTION)
000000 RIV0ALEN DS    X                    VECTOR LENGTH
000001 RIV0AKEY DS    X                    VECTOR KEY
       RIV0AKYC EQU   X'0A'                KEY IS X'0A'
000002 RIV0ADTA DS    X                    VECTOR DATA
       *
```

```
       ************************************************************************
       *** ISTRIV0B - maps the LU 6.2 application definition vector.        **
       ***            After the LU 6.2 application program has issued an    **
       ***            open ACB, the LU 6.2 application program may use      **
       ***            this vector to determine the values coded on the      **
       ***            APPL definition statement.                            **
       ***          - This is obtained from the APPL STATEMENT PARAMETERS   **
       ************************************************************************
000000 ISTRIV0B DSECT                      LU 6.2 APPL DEFINITION VECTOR
       *                                   (FROM APPL STATEMENT PARAMETERS)
000000 RIV0BLEN DS    X                    VECTOR LENGTH
000001 RIV0BKEY DS    X                    VECTOR KEY
       RIV0BKYC EQU   X'0B'                KEY IS X'0B'
000002 RIV0BDTA DS    0X                   VECTOR DATA
000002          DS    X                    RESERVED
       RIV0BSLV EQU   X'C0'                SESSION-LEVEL LU-LU VERIFICATION
       *                                   BIT MASK
       RIV0BSLR EQU   X'80'                REQUIRED
       RIV0BSLO EQU   X'40'                OPTIONAL
       RIV0BSLN EQU   X'00'                NONE
000003 RIV0BCLS DS    X                    CONVERSATION SECURITY ACCEPTANCE
       RIV0BCLN EQU   X'01'                NONE
       RIV0BCLC EQU   X'02'                CONV
       RIV0BCLA EQU   X'03'                ALREADYV
       RIV0BCLP EQU   X'04'                PERSISTV
       RIV0BCLV EQU   X'05'                AVPV
000004 RIV0BFLG DS    X                    MISCELLANEOUS FLAGS
       RIV0BDDL EQU   X'80'                DDRAINL=ALLOW
       RIV0BDRL EQU   X'40'                DRESPL=ALLOW
       RIV0BATA EQU   X'20'                ATNLOSS=ALL
       RIV0BSYP EQU   X'10'                SYNCLVL=SYNCPT
       RIV0BOPC EQU   X'08'                OPERCNOS=ALLOW
000005          DS    X                    RESERVED
000006 RIV0BDSL DS    HL2                  DSESLIM VALUE
000008 RIV0BDML DS    HL2                  DMINWNL VALUE
00000A RIV0BDMR DS    HL2                  DMINWNR VALUE
00000C RIV0BAUT DS    HL2                  AUTOSES VALUE
       *
       ************************************************************************
       *** ISTRIV0C - maps the common application definition vector.        **
       ***            After the application program has issued an open for  **
       ***            its ACB, the application may examine this vector to   **
       ***            determine the values coded on the APPL definition     **
       ***            statement for common application defination keywords. **
       ***          - This is obtained from the APPL STATEMENT PARAMETERS   **
       ************************************************************************
000000 ISTRIV0C DSECT                      APPLICATION DEFINITION VECTOR
       *                                   FOR ALL APPLICATION PROGRAMS   @N1A
       *                                   (FROM APPL STATEMENT PARAMETERS)
000000 RIV0CLEN DS    X                    VECTOR LENGTH                  @N1A
000001 RIV0CKEY DS    X                    VECTOR KEY                     @N1A
       RIV0CKYC EQU   X'0C'                KEY IS X'0C'                   @N1A
000002 RIV0CDTA DS    0X                   VECTOR DATA                    @N1A
000002 RIV0CAUT DS    X                    AUTHORIZATION SETTINGS         @N1A
       RIV0CACQ EQU   X'80'                AUTH=ACQ                       @N1A
       RIV0CASD EQU   X'40'                AUTH=ASDP                      @N1A
       RIV0CCNM EQU   X'20'                AUTH=CNM                       @N1A
       RIV0CPAS EQU   X'10'                AUTH=PASS                      @N1A
       RIV0CPPO EQU   X'08'                AUTH=PPO                       @N1A
       RIV0CSPO EQU   X'04'                AUTH=SPO                       @N1A
       RIV0CTSO EQU   X'02'                AUTH=TSO                       @N1A
       RIV0CVPA EQU   X'01'                AUTH=VPACE                     @N1A
000003 RIV0CFL1 DS    X                    MISCELLANEOUS FLAGS 1          @N1A
       RIV0CAPC EQU   X'80'                APPC=YES                       @N1A
       RIV0CAUX EQU   X'40'                AUTHEXIT=YES                   @N1A
       RIV0CCER EQU   X'20'                CERTIFY=YES                    @N1A
       RIV0CDSW EQU   X'10'                DSPLYWLD=YES                   @N1A
```

```
       RIV0CFSP EQU   X'08'                FASTPASS=YES                   @N1A
       RIV0CHAV EQU   X'04'                HAVAIL=YES                     @N1A
       RIV0CPAR EQU   X'02'                PARSESS=YES                    @N1A
       RIV0CPRS EQU   X'01'                PERSIST=MULTI                  @N1A
000004 RIV0CFL2 DS    X                    MISCELLANEOUS FLAGS 2          @N1A
       RIV0CSSL EQU   X'80'                SESSLIM=YES                    @N1A
       RIV0CSON EQU   X'40'                SONSCIP=YES                    @N1A
       RIV0CSRX EQU   X'20'                SRBEXIT=YES                    @N1A
       RIV0CVCN EQU   X'10'                VCNS=YES                       @N1A
       RIV0CVFR EQU   X'08'                VTAMFRR=YES                    @N1A
000005 RIV0CLTM DS    X                    LOSTERM SETTING                @N1A
       RIV0CLTN EQU   X'00'                LOSTERM=NORMAL                 @N1A
       RIV0CLTI EQU   X'01'                LOSTERM=IMMED                  @N1A
       RIV0CLTS EQU   X'02'                LOSTERM=SECOND                 @N1A
000006 RIV0CCMI DS    X                    CMPAPPLI VALUE                 @N1A
000007 RIV0CCMO DS    X                    CMPAPPLO VALUE                 @N1A
000008 RIV0CENC DS    X                    ENCR VALUE                     @N1A
       RIV0CECN EQU   X'00'                ENCR=NONE                      @N1A
       RIV0CECO EQU   X'01'                ENCR=OPT                       @N1A
       RIV0CECC EQU   X'02'                ENCR=COND                      @N1A
       RIV0CECS EQU   X'03'                ENCR=SEL                       @N1A
       RIV0CECR EQU   X'04'                ENCR=REQD                      @N1A
000009 RIV0CVPC DS    X                    VPACING VALUE                  @N1A
00000A          DS    XL4                  RESERVED                       @N1A
       *
       ************************************************************************
       *** ISTRIV11 - maps the APPCCMD vector area length vector.           **
       ***          - It contains the absolute minimum length and the       **
       ***            recommended minimum length for full use of the        **
       ***            APPCCMD vector area.                                  **
       ************************************************************************
000000 ISTRIV11 DSECT                      APPCCMD VECTOR AREA LENGTH VECTOR
       *                                                                 @L3C
000000 RIV11LEN DS    X                    VECTOR LENGTH
000001 RIV11KEY DS    X                    VECTOR KEY
       RIV11KYC EQU   X'11'                KEY IS X'11'
000002 RIV11AML DS    XL4                  ABSOLUTE MINIMUM APPCCMD VECTOR
       *                                   AREA LENGTH                 @L3A
000006 RIV11RML DS    XL4                  RECOMMENDED MINIMUM APPCCMD
       *                                   VECTOR AREA LENGTH          @L3C
       *
       ************************************************************************
       *** ISTRIV12 - maps the application to VTAM vector keys vector.      **
       ***          - It contains a list of all ACB vector keys that        **
       ***            VTAM will process. Constants for the ACB vectors are  **
       ***            located in ISTVACBV.                                  **
       ************************************************************************
000000 ISTRIV12 DSECT                      APPLICATION TO VTAM VECTOR KEYS
       *                                   FOR ACB MACRO
000000 RIV12LEN DS    X                    VECTOR LENGTH
000001 RIV12KEY DS    X                    VECTOR KEY
       RIV12KYC EQU   X'12'                KEY IS X'12'
000002 RIV12DTA DS    0CL1                 VECTOR DATA
       *
       ************************************************************************
       *** ISTRIV13 - maps the Performance Monitor vector.                  **
       ***            Identifies a table of Performance Data vector fields  **
       ***            (within ISTXPL) that have been retired by the         **
       ***            Performance Monitor Interface since its inception.    **
       ************************************************************************
000000 ISTRIV13 DSECT                      PERFORMANCE MONITOR VECTOR   @L1A
       *
000000 RIV13LEN DS    X                    VECTOR LENGTH                @L1A
000001 RIV13KEY DS    X                    VECTOR KEY                   @L1A
       RIV13KYC EQU   X'13'                KEY IS X'13'                 @L1A
000002 RIV13ENT DS    HL2                  NUMBER OF ENTRIES IN TABLE   @L2A
       *                                   (ZERO IF NONE RETIRED)
```

```
000004 RIV13RFT DS    AL4                 RETIRED FIELDS TABLE ADDRESS @L2A
       *                                  (ZERO IF NONE RETIRED)
000008 RIV13ELN DS    HL2                 LENGTH OF EACH ENTRY         @L2A
       *
000000 RIV13TBL DSECT                     RETIRED FIELDS TABLE ENTRY   @L1A
       *                                  (MAPS ENTRIES IN TABLE ADDRESSED
       *                                  BY RIV13RFT)
000000 RIV13VID DS    0CL6                ID OF AFFECTED VECTOR        @L1A
000000 RIV13MAJ DS    CL2                 MAJOR CATEGORY               @L1A
000002 RIV13SUB DS    CL2                 SUBCATEGORY                  @L1A
000004 RIV13REC DS    CL2                 RECORD TYPE                  @L1A
000006 RIV13FLD DS    0CL4                FIELD POSITION WITHIN VECTOR @L1A
000006 RIV13OFF DS    HL2                 FIELD OFFSET                 @L1A
000008 RIV13FLG DS    BL1                 FLAG BYTE                    @L1A
       RIV13BIT EQU   X'01'               DATA TYPE INDICATOR
       *                                  (1= BITSTRING, 0= OTHER)     @L1A
000009 RIV13LNG DS    XL1                 FIELD LENGTH IF NOT BITSTRING,
       *                                  MASK FOR BITS RETIRED WITHIN BYTE
       *                                  FOR BITSTRING FIELD
       *
```

Extended Buffer List Entry (ISTBLXEN)

```
  LOC    SOURCE STATEMENT
000000 ISTBLXEN DSECT
000000 BLXEN_CSM       DS    0CL28          THIS AREA MAPS THE CSM
       *                                   BUFFER DESCRIPTOR
000000                 DS    X              RESERVED.  THIS FIELD MUST BE
       *                                   SET TO ZERO.
000001 BLXEN_SOURCE    DS    X              BUFFER SOURCE
       BLXEN_CECSA     EQU   X'80'          INDICATES THAT THE STORAGE
       *                                   REFERENCED IN THE LIST IS
       *                                   CSM ECSA
       BLXEN_CDSPACE   EQU   X'40'          INDICATES THAT THE STORAGE
       *                                   REFERENCED IN THE LIST IS
       *                                   CSM DATA SPACE
000002 BLXEN_TYPE      DS    X              BUFFER TYPE
       BLXEN_FIXED     EQU   X'80'          INDICATES THAT THE STORAGE IS
       *                                   IN A GUARANTEED TO BE FIXED
       *                                   STATE
       BLXEN_PAGEABLE  EQU   X'40'          INDICATES THAT THE STORAGE IS
       *                                   IN A GUARANTEED TO BE PAGEABLE
       *                                   STATE
       BLXEN_PAGEELIG  EQU   X'20'          INDICATES THAT THE STORAGE
       *                                   IS ELIGIBLE TO BE PAGEFREED BY
       *                                   CSM
000003                 DS    XL1            RESERVED
000004 BLXEN_CTOKN     DS    XL12           CSM TOKEN
000010 BLXEN_ALET      DS    F              CSM DATA SPACE ALET
000014 BLXEN_AREA      DS    A              POINTER TO DATA
000018 BLXEN_RLEN      DS    F              LENGTH OF DATA
00001C BLXEN_RLENA     DS    F              LENGTH OF DATA ACCEPTED BY
       *                                   ... VTAM ON A REQUEST TO SEND
       *                                   ... DATA.
       *                                   ... THIS FIELD SHOULD BE SET
       *                                   ... TO ZERO BY THE APPLICATION.
       *                                   ... VTAM SETS THIS FIELD
       *                                   ... TO REFLECT THE AMOUNT OF
       *                                   ... DATA REFERENCED BY XBUFLST
       *                                   ... THAT HAS BEEN ACCEPTED BY
       *                                   ... VTAM.
000020 BLXEN_VAFLAGS   DS    X              VTAM and APPL FLAGS
       BLXEN_OWNACC    EQU   X'80'          VTAM HAS ACCEPTED OWNERSHIP
       *                                   OF THE CSM BUFFER
000021                 DS    XL15           RESERVED
```

Chapter 4. Summary of Register Usage

Table 3 shows what VTAM does with the general-purpose registers before it returns control to the application program at the next sequential instruction. It indicates which registers are left unchanged by the VTAM macroinstructions and which ones can be modified between the time the macroinstruction is executed and control is returned to the application program. The table also shows the disposition of the registers when any of the exit routines receive control. Refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* for further details on how to handle macroinstruction errors.

Table 3. Register Contents upon Return of Control

	Register 0	**Register 1**	**Register 2-12**	**Register 13**	**Register 14**	**Register 15**
Upon return from OPEN and CLOSE macroinstructions	Unpredictable	Unpredictable	Unmodified	Unmodified [1]	Unpredictable	Return code
Upon return from RPL-based macroinstructions, including CHECK	See footnote [2]	Address of RPL	Unmodified	Unmodified [1]	Unpredictable	See footnote [1]
Upon return from GENCB	Error return code or control block length [3] [4]	Control block address [3] [4]	Unmodified	Unmodified [1]	Unpredictable	General return code
Upon return from SHOWCB, MODCB, or TESTCB	Error return code [4]	Unpredictable	Unmodified	Unmodified [1]	Unpredictable	General return code
Upon invocation of LERAD or SYNAD exit routines	Recovery action return code	Address of RPL	Unmodified	Unmodified [1]	Return address	Address of exit routine
Upon invocation of other EXLST exit routines	Unpredictable	Address of VTAM-supplied parameter list	Unpredictable	Unpredictable	Return address	Address of exit routine

1. Register 13 must indicate the address of an 18-word save area when the macroinstruction is executed.
2. If the operation completed normally, register 15 is set to 0. For some macroinstructions completing normally but with a special condition, register 0 is also set. If an error occurred and the LERAD or SYNAD exit routine has been invoked, registers 0 and 15 contain the values set in them by the exit routine. If an error occurred and no LERAD or SYNAD exit routine exists, VTAM sets register 15 to 4 and places a recovery action return code in register 0 (if the error is that the ACB is not open, register 15 is set to decimal 32 and the RPL request code is set in register 0).
3. When GENCB completes successfully (register 15 is set to 0), register 1 contains the address of the generated control blocks and register 0 contains the length of the control blocks, in bytes.
4. If GENCB, SHOWCB, MODCB, or TESTCB completes unsuccessfully (with register 15 not set to 0), register 1 is unpredictable and register 0 contains an error code (if register 15 is set to 4 or 12) or else is unpredictable.

© Copyright IBM Corp. 1988, 2001

Summary of Register Usage

Table 3. Register Contents upon Return of Control (continued)

	Register 0	Register 1	Register 2-12	Register 13	Register 14	Register 15
Upon invocation of RPL-based exit routines	Unpredictable	Address of RPL	Unpredictable	Unpredictable	Return address	Address of exit routine

Appendix A. Information Apars

This appendix lists information apars for IP and SNA books.

Notes:

1. Information apars contain updates to previous editions of the manuals listed below. Books updated for V1R2 are complete except for the updates contained in the information apars that may be issued after V1R2 books went to press.
2. Information apars are predefined for z/OS V1R2 Communications Server and may not contain updates.

IP Information Apars

Table 4 lists information apars for IP books.

Table 4. IP Information Apars

Title	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8	CS for OS/990 2.7	CS for OS/390 2.6	CS for OS/390 2.5
IP API Guide	ii12861	ii12371	ii11635	ii11558	ii11405	ii11144
IP CICS Sockets Guide	ii12862		ii11626	ii11559	ii11406	ii11145
IP Configuration			ii11620 ii12068 ii12353 ii12649	ii11555 ii11637 ii11995 ii12325	ii11402 ii11619 ii12066 ii12455	ii11159 ii11979 ii12315
IP Configuration Guide	ii12498	ii12362 ii12493				
IP Configuration Reference	ii12499	ii12363 ii12494 ii12712				
IP Diagnosis	ii12503	ii12366 ii12495	ii11628	ii11565	ii11411	ii11160 ii11414
IP Messages Volume 1	ii12857	ii12367	ii11630	ii11562	ii11408	ii11636
IP Messages Volume 2	ii12858	ii12368	ii11631	ii11563	ii11409	ii11281
IP Messages Volume 3	ii12859	ii12369	ii11632 ii12883	ii11564 ii12884	ii11410 ii12885	ii11158
IP Messages Volume 4	ii12860					
IP Migration	ii12497	ii12361	ii11618	ii11554	ii11401	ii11204
IP Network Print Facility	ii12864		ii11627	ii11561	ii11407	ii11150
IP Programmer's Reference	ii12505		ii11634	ii11557	ii11404	ii12496

© Copyright IBM Corp. 1988, 2001

Table 4. IP Information Apars (continued)

Title	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8	CS for OS/990 2.7	CS for OS/390 2.6	CS for OS/390 2.5
IP and SNA Codes	ii12504	ii12370	ii11917	Added TCP/IP codes to VTAM codes V2R6 ii11611	ii11361	ii11146 ii11097
IP User's Guide		ii12365	ii11625	ii11556	ii11403	ii11143
IP User's Guide and Commands	ii12501					
IP System Admin Guide	ii12502					
Quick Reference	ii12500	ii12364				

SNA Information Apars

Table 5 lists information apars for SNA books.

Table 5. SNA Information Apars

Title	z/OS CS V1R2	CS for OS/390 2.10 and z/OS CS V1R1	CS for OS/390 2.8	CS for OS/390 2.7	CS for OS/390 2.6	CS for OS/390 2.5
Anynet SNA over TCP/IP			ii11922	ii11633	ii11624	ii11623
Anynet Sockets over SNA			ii11921	ii11622	ii11519	ii11518
CSM Guide						
IP and SNA Codes		ii12370	ii11917	ii11611	ii11361	ii11097
SNA Customization	ii12872	ii12388	ii11923	ii11925 ii12008	ii11924 ii12007	ii11092 ii11621 ii12006
SNA Diagnosis	ii12490	ii12389	ii11915	ii11615	ii11357	ii11585
SNA Messages	ii12491	ii12382	ii11916	ii11610	ii11358	ii11096
SNA Network Implementation Guide	ii12487	ii12381	ii11911	ii11609 ii12683	ii11353 ii11493	ii11095
SNA Operation	ii12489	ii12384	ii11914	ii11612	ii11355	ii11098
SNA Migration	ii12486	ii12386	ii11910	ii11614	ii11359	ii11100
SNA Programming		ii12385	ii11920	ii11613	ii11360	ii11099
Quick Reference	ii12500	ii12364	ii11913	ii11616	ii11356	
SNA Resource Definition Reference	ii12488	ii12380 ii12567	ii11912 ii12568	ii11608 ii12569	ii11354 ii12259 ii12570	ii11094 ii11151 ii12260 ii12571
SNA Resource Definition Samples						

Appendix B. Notices

IBM may not offer all of the products, services, or features discussed in this document. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

© Copyright IBM Corp. 1988, 2001

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O.Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly

tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

This product includes cryptographic software written by Eric Young.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats of unlicensed books and the z/OS Licensed Product Library (LK3T-4307), which contains BookManager and PDF formats of licensed books.

Programming Interface Information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of z/OS Communications Server.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
Advanced Peer-to-Peer Networking
AFP
AD/Cycle
AIX
AIX/ESA
AnyNet
APL2
APPN
AS/400
AT
BookManager
BookMaster
CBPDO
C/370
CICS
CICS/ESA
C/MVS
Common User Access
C Set ++
CT
CUA
DATABASE 2
DatagLANce
DB2
DFSMS
DFSMSdfp
DFSMShsm
DFSMS/MVS
DPI
Domino
DRDA
eNetwork
Enterprise Systems Architecture/370
ESA/390
ESCON
@server
ES/3090
ES/9000
ES/9370
EtherStreamer
Extended Services
FAA
Micro Channel
MVS
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
MQ
Natural
NetView
Network Station
Nways
Notes
NTune
NTuneNCP
OfficeVision/MVS
OfficeVision/VM
Open Class
OpenEdition
OS/2
OS/390
OS/400
Parallel Sysplex
Personal System/2
PR/SM
PROFS
PS/2
RACF
Resource Measurement Facility
RETAIN
RFM
RISC System/6000
RMF
RS/6000
S/370
S/390
SAA
SecureWay
Slate
SP
SP2
SQL/DS
System/360

FFST
FFST/2
FFST/MVS
First Failure Support Technology
GDDM
Hardware Configuration Definition
IBM
IBMLink
IMS
IMS/ESA
InfoPrint
Language Environment
LANStreamer
Library Reader
LPDA
MCS
System/370
System/390
SystemView
Tivoli
TURBOWAYS
UNIX System Services
Virtual Machine/Extended Architecture
VM/ESA
VM/XA
VSE/ESA
VTAM
WebSphere
XT
z/Architecture
z/OS
zSeries
400
3090
3890

Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

DB2 and NetView are registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the U.S., other countries, or both.

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see http://www.intel.com/tradmarx.htm.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

AAREA keyword
- accepting a session 146
- allocation request, conditional, without wait 56
- allocation request for specific session 21
- allocation requests for contention-winner session 33
- allocation requests that can be queued 9
- any-mode RECEIVEs 345
- as pointer to RPL extension 392
- CNOS requests 152
- conditional deallocation 99
- conditional deallocation that includes data 108
- deallocation due to program errors 70
- deallocation due to service errors 78
- deallocation due to timer errors 85
- deallocation due to user-specified criteria 92
- defining LU-mode table values 167
- displaying LU-mode table values 173
- entering receive state conditionally 243
- entering receive state conditionally and including data 252
- entering receive state unconditionally 275
- entering receive state unconditionally and including data 264
- flushing the buffer 484
- immediate allocation request 45
- indicating the beginning of a synchronization exchange 538
- indicating the end of a synchronization exchange 542
- obtain status on information available on a specified conversation 567
- obtain status on information from any active conversation 547
- obtain status on information immediately available from any active conversation 554
- obtain status on information immediately available on a specified conversation 560
- queued deallocation due to program, service, timer, or user errors 138
- receive normal information immediately available from any conversation 357
- receive normal information immediately available on a specific conversation 368
- receiving an FMH-5 327
- receiving expedited data from the specified conversation 306
- receiving expedited data immediately available in any-mode 290
- receiving expedited data immediately from the specified conversation 298
- receiving expedited data in any-mode 282
- rejecting a session 161
- releasing a suspended session 527
- replying to a confirmation request 425
- requesting permission to enter send state 492
- resetting the continuation mode 410

AAREA keyword *(continued)*
- restoring a mode 181
- sending an error notification 470
- sending data 433
- sending data and flushing the buffer 458
- sending data conditionally 417, 446
- sending expedited data on a full-duplex session 499
- specific-mode RECEIVEs 380
- suspending a subsequent conversation 533
- terminating a session 392
- terminating a session due to errors 404
- terminating session due to protocol error or cleanup deactivation 398
- unconditional deallocation 130
- unconditional deallocation that includes data 120

abnormal conversation deallocation
- LU Services errors (ABNDSERV) 76, 136
- reference material (macroinstruction syntax and operands) 68, 90
- timing errors (ABNDTIME) 83, 136
- transaction program errors (ABNDPROG) 68, 136
- user-defined errors (ABNDUSER) 90, 136

ACB keyword
- accepting a session 146
- allocation request, conditional, without wait 56
- allocation request for contention-winner session 33
- allocation request for specific session 21
- allocation requests that can be queued 9
- any-mode RECEIVEs 345
- CNOS requests 152
- conditional deallocation 99
- conditional deallocation that includes data 108
- deallocation due to program errors 70
- deallocation due to service errors 78
- deallocation due to timer errors 85
- deallocation due to user-specified criteria 92
- defining LU-mode table values 167
- displaying LU-mode table values 174
- entering receive state conditionally 243
- entering receive state conditionally and including data 252
- entering receive state unconditionally 275
- entering receive state unconditionally and including data 264
- flushing the buffer 484
- immediate allocation request 45
- indicating the beginning of a synchronization exchange 538
- indicating the end of a synchronization exchange 542
- obtain status on information available on a specified conversation 567
- obtain status on information from any active conversation 548
- obtain status on information immediately available from any active conversation 554

© Copyright IBM Corp. 1988, 2001

ACB keyword *(continued)*
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available from any conversation 357
receive normal information immediately available on a specific conversation 368
receiving an FMH-5 327
receiving expedited data from the specified conversation 306
receiving expedited data immediately available in any-mode 290
receiving expedited data immediately from the specified conversation 298
receiving expedited data in any-mode 282
rejecting a session 161
releasing a suspended session 528
replying to a confirmation request 425
requesting permission to enter send state 492
resetting the continuation mode 410
restoring a mode 181
sending an error notification 470
sending data 433
sending data and flushing the buffer 458
sending data conditionally 417, 446
sending expedited data on a full-duplex session 500
specific-mode RECEIVEs 380
suspending a subsequent conversation 533
terminating a session 392
terminating a session due to errors 404
terminating a session due to protocol errors or cleanup deactivation 398
unconditional deallocation 130
unconditional deallocation that includes data 120
accepting a session, macroinstruction 144
accepting a session macroinstruction 144
access method control block (ACB) keyword
accepting a session 146
allocation request, conditional, without wait 56
allocation request for contention-winner session 33
allocation request for specific session 21
allocation requests that can be queued 9
any-mode RECEIVEs 345
CNOS requests 152
conditional deallocation 99
conditional deallocation that includes data 108
deallocation due to program errors 70
deallocation due to service errors 78
deallocation due to timer errors 85
deallocation due to user-specified criteria 92
defining LU-mode table values 167
displaying LU-mode table values 174
entering receive state conditionally 243
entering receive state conditionally and including data 252
entering receive state unconditionally 275
entering receive state unconditionally and including data 264
access method control block (ACB) keyword *(continued)*
flushing the buffer 484
immediate allocation request 45
indicating the beginning of a synchronization exchange 538
indicating the end of a synchronization exchange 542
obtain status on information available on a specified conversation 567
obtain status on information from any active conversation 548
obtain status on information immediately available from any active conversation 554
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available from any conversation 357
receive normal information immediately available on a specific conversation 368
receiving an FMH-5 327
receiving expedited data from the specified conversation 306
receiving expedited data immediately available in any-mode 290
receiving expedited data immediately from the specified conversation 298
receiving expedited data in any-mode 282
rejecting a session 161
releasing a suspended session 528
replying to a confirmation request 425
requesting permission to enter send state 492
resetting the continuation mode 410
restoring a mode 181
sending an error notification 470
sending data 433
sending data and flushing the buffer 458
sending data conditionally 417, 446
sending expedited data on a full-duplex session 500
specific-mode RECEIVEs 380
suspending a subsequent conversation 533
terminating a session 392
terminating a session due to errors 404
terminating a session due to protocol errors or cleanup deactivation 398
unconditional deallocation 130
unconditional deallocation that includes data 120
activating a session, macroinstruction for 144
ACTSESS qualify value 144
allocating a conversation
APPCCMD CONTROL=ALLOC macroinstruction macroinstruction syntax and operands 7, 43
session assignment algorithm
conditional request, without wait 54
immediate requests 43
request for specific session 19
requests for contention-winner sessions 31
requests that can be queued 7

ALLOCD qualify value
  session assignment algorithm 7
  syntax and operands 7
any-mode RECEIVEs, macroinstruction 343
APPCCMD macroinstruction
  coding comments 3
  continuation lines 3
AREA keyword,
  as pointer to
    DEFINE/DISPLAY control block 174
    session limits control block 152
  returned value, used with
    obtain status on information available on a specified conversation 567
    obtain status on information from any active conversation 548
    obtain status on information immediately available from any active conversation 554
    obtain status on information immediately available on a specified conversation 560
  used with
    accepting a session 146
    allocation request, conditional, without wait 56
    allocation request for contention-winner session 34
    allocation request for specific session 21
    allocation request that can be queued 10
    any-mode RECEIVEs 346
    CNOS requests 152
    conditional deallocation that includes data 108
    deallocation due to program errors 70
    deallocation due to service errors 78
    deallocation due to timer errors 85
    deallocation due to user-specified criteria 92
    defining LU-mode table values 167
    displaying LU-mode table values 174
    entering receive state conditionally and including data 252
    entering receive state unconditionally and including data 264
    immediate allocation request 45
    queued deallocation due to program, service, timer, or user errors 139
    receive normal information immediately available from any conversation 357
    receive normal information immediately available on a specific conversation 368
    receiving an FMH-5 327
    receiving expedited data from the specified conversation 306
    receiving expedited data immediately available in any-mode 290
    receiving expedited data immediately from the specified conversation 298
    receiving expedited data in any-mode 282
    restoring a mode 181
    sending an error notification 471
    sending data 433
    sending data and flushing the buffer 458
    sending data conditionally 446
AREA keyword, *(continued)*
  used with *(continued)*
    sending expedited data on a full-duplex session 500
    specific-mode RECEIVEs 380
    unconditional deallocation that includes data 120
AREALEN keyword, used with
  any-mode RECEIVEs 346
  displaying LU-mode table values 174
  obtain status on information available on a specified conversation 567
  obtain status on information from any active conversation 548
  obtain status on information immediately available from any active conversation 554
  obtain status on information immediately available on a specified conversation 560
  receive normal information immediately available from any conversation 357
  receive normal information immediately available on a specific conversation 369
  receiving an FMH-5 327
  receiving expedited data from the specified conversation 306
  receiving expedited data immediately available in any-mode 290
  receiving expedited data immediately from the specified conversation 298
  receiving expedited data in any-mode 282
  restoring a mode 181
  specific-mode RECEIVEs 381
ARG keyword
  relationship to CID 161
  used with accepting a session 146
  used with rejecting a session 161
AVFA RPL field
  returned value, used with
    allocation request, conditional, without wait 61
    allocation request for contention-winner session 38
    allocation request for specific session 26
    allocation request that can be queued 14
    CNOS request 156
    immediate allocation request 49

B

becoming receiving LU 241, 274
becoming sender
  requesting permission to enter send state macroinstruction 490
BIND image
  DSECT 625
  relationship to
    RESTORE 179
BNDAREA DSECT
  profile 1 628
  profile 2 631
  profile 3 632
  profile 4 632
  profile 6 634

BRANCH keyword, used with
accepting a session 146
allocation request, conditional, without wait 57
allocation request for contention-winner session 34
allocation request for specific session 22
allocation request that can be queued 10
any-mode RECEIVEs 346
CNOS requests 152
conditional deallocation 100
conditional deallocation that includes data 108
deallocation due to program errors 71
deallocation due to service errors 78
deallocation due to timer errors 85
deallocation due to user-specified criteria 93
defining LU-mode table values 167
displaying LU-mode table values 174
entering receive state conditionally 243
entering receive state conditionally and including data 252
entering receive state unconditionally 275
entering receive state unconditionally and including data 264
flushing the buffer 484
immediate allocation request 45
indicating the beginning of a synchronization exchange 538
indicating the end of a synchronization exchange 543
obtain status on information available on a specified conversation 567
obtain status on information from any active conversation 548
obtain status on information immediately available from any active conversation 554
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available from any conversation 357
receive normal information immediately available on a specific conversation 369
receiving an FMH-5 327
receiving expedited data from the specified conversation 306
receiving expedited data immediately available in any-mode 290
receiving expedited data immediately from the specified conversation 298
receiving expedited data in any-mode 282
rejecting a session 162
releasing a suspended session 528
replying to a confirmation request 426
requesting permission to enter send state 492
resetting the continuation mode 410
restoring a mode 181
sending an error notification 471
sending data 434
sending data and flushing the buffer 458
sending data conditionally 417, 446
BRANCH keyword, used with *(continued)*
sending expedited data on a full-duplex session 500
specific-mode RECEIVEs 381
terminating a session 392
terminating a session due to errors 404
terminating a session due to protocol errors or cleanup deactivation 398
unconditional deallocation 131
unconditional deallocation that includes data 120
used with suspending a subsequent conversation 533

C

canceling APPCCMD macroinstructions
abnormal deallocation (syntax and operands) 68, 90
terminating a session (syntax and operands) 402
CD keyword, used with
any-mode RECEIVEs 346
ISTRPL6 577
receive normal information immediately available from any conversation 358
receive normal information immediately available on a specific conversation 369
specific-mode RECEIVEs 381
CGID
keyword, used with
allocation request for specific session 22
terminating a session due to protocol errors or cleanup deactivation 398
CGID RPL field
returned value, used with
allocation request, conditional, without wait 61
allocation request for contention-winner session 38
allocation request that can be queued 14
immediate allocation request 49
receiving an FMH-5 330
change number of sessions (CNOS) macroinstruction 150
changing continuation modes
use of RESETRCV macroinstruction variation 408
CHECK control value 66
CHECK macroinstruction 66
CNOS (change number of sessions) macroinstruction 150
CNOS session limits data structure—ISTSLCNS DSECT 645
coding comments 2, 4
CONFIRM qualify value
DEALLOC CONTROL value 98
PREPRCV CONTROL value 241
SEND CONTROL value 415
confirmation request
replying to a confirmation request macroinstruction, used with 424
sending an error notification macroinstruction, used with 468
CONFRMD qualify value 424

CONFTXT keyword, used with
accepting a session 147
CONMODE keyword, used with
allocation request, conditional, without wait 57
allocation request for contention-winner session 34
allocation request for specific session 22
allocation request that can be queued 10
any-mode RECEIVES 347
conditional deallocation 100
conditional deallocation that includes data 109, 121, 131
entering receive state conditionally 243
entering receive state conditionally and including data 253
entering receive state unconditionally 276
entering receive state unconditionally and including data 265
flushing the buffer 484
immediate allocation request 46
ISTRPL6 577
receive normal information immediately available from any conversation 358
receive normal information immediately available on a specific conversation 369
receiving expedited data from the specified conversation 307
receiving expedited data immediately available in any-mode 290
receiving expedited data immediately from the specified conversation 299
receiving expedited data in any-mode 282
replying to a confirmation request 426
requesting permission to enter send state 492
resetting the continuation mode 410
sending an error notification 471
sending data 434
sending data and flushing the buffer 459
sending data conditionally 417, 447
sending expedited data on a full-duplex session 500
specific-mode RECEIVEs 381
CONSTATE RPL field
returned value, used with
allocation request, conditional, without wait 61
allocation request for contention-winner session 38
allocation request for specific session 26
allocation request that can be queued 14
any-mode RECEIVEs 349
conditional deallocation 102
conditional deallocation that includes data 112
deallocation due to program errors 72
deallocation due to service errors 80
deallocation due to timer errors 87
deallocation due to user-specified criteria 94
entering receive state conditionally 246
entering receive state conditionally and including data 256
entering receive state unconditionally 278
entering receive state unconditionally and including data 268
CONSTATE RPL field *(continued)*
returned value, used with *(continued)*
flushing the buffer 487
immediate allocation request 49
obtain status on information available on a specified conversation 569
obtain status on information immediately available on a specified conversation 562
queued deallocation due to program, service, timer, or user errors 141
receive normal information immediately available from any conversation 360
receive normal information immediately available on a specific conversation 372
receiving an FMH-5 330
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 292
receiving expedited data immediately from the specified conversation 301
receiving expedited data in any-mode 284
replying to confirmation request 428
requesting permission to enter send state 494
resetting the continuation mode 412
sending an error notification 474
sending data 438
sending data and flushing the buffer 462
sending data conditionally 419, 450
sending expedited data on full-duplex session 503
specific-mode RECEIVEs 384
terminating a session 394
unconditional deallocation 133
unconditional deallocation that includes data 124
contention winner
requests for allocation 31
continuation lines 2, 4
CONTROL keyword
ALLOC value
ALLOCD QUALIFY value 7
CONVGRP QUALIFY value 19
CONWIN QUALIFY value 31
IMMED QUALIFY value 43
WHENFREE QUALIFY value 54, 230
CHECK value 66
DEALLOC value
ABNDPROG QUALIFY value 68
ABNDSERV QUALIFY value 76
ABNDTIME QUALIFY value 83
ABNDUSER QUALIFY value 90
CONFIRM QUALIFY value 98
DATACON QUALIFY value 106
DATAFLU QUALIFY value 118
FLUSH QUALIFY value 129
DEALLOCQ value
ABNDPROG QUALIFY value 136
ABNDSERV QUALIFY value 136
ABNDTIME QUALIFY value 136
ABNDUSER QUALIFY value 136

CONTROL keyword *(continued)*
OPRCNTL value
ACTSESS QUALIFY value 144
CNOS QUALIFY value 150
DACTSESS QUALIFY value 160
DEFINE QUALIFY value 165
DISPLAY QUALIFY value 172
RESTORE QUALIFY value 179
PREPRCV value
CONFIRM QUALIFY value 241
DATACON QUALIFY value 250
DATAFLU QUALIFY value 262
FLUSH QUALIFY value 274
RCVEXPD value
ANY QUALIFY value 280
IANY QUALIFY value 288
ISPEC QUALIFY value 296
SPEC QUALIFY value 304
RCVFMH5 value 325
RECEIVE value
ANY QUALIFY value 343
IANY QUALIFY value 355
ISPEC QUALIFY value 366
SPEC QUALIFY value 378
REJECT value
CONV QUALIFY value 390
CONVGRP QUALIFY value 396
SESSION QUALIFY value 402
RESETRCV value 408
SEND value
CONFIRM QUALIFY value 415
CONFRMD QUALIFY value 424
DATA QUALIFY value 431
DATACON QUALIFY value 444
DATAFLU QUALIFY value 456
ERROR QUALIFY value 468
FLUSH QUALIFY value 482
RQSEND QUALIFY value 490
SENDEXPD value 497
SETSESS value
RESUME QUALIFY value 526
SUSPEND QUALIFY value 531
SYNCBEG QUALIFY value 536
SYNCEND QUALIFY value 541
TESTSTAT value
ALL QUALIFY value 546
IALL QUALIFY value 552
ISPEC QUALIFY value 558
SPEC QUALIFY value 565
conversation allocation
APPCCMD CONTROL=ALLOC macroinstruction
macroinstruction syntax and operands 7, 43
session assignment algorithm
conditional request, without wait 54, 230
immediate requests 43
request for specific session 19
requests for contention-winner sessions 31
requests that can be queued 7
conversation data, receiving
APPCCMD macroinstructions used 343, 378
conversation data, sending
APPCCMD macroinstructions used
syntax and operands 415, 490
buffer list (OPTCD=BUFFLST) considerations, used with
conditional deallocation that includes data 110
entering receive state conditionally and including data 255
sending data 112, 436
sending data and flushing the buffer 255, 267, 461
sending data conditionally 449
sending expedited data on a full-duplex session 502
unconditional deallocation that includes data 124
flushing the buffer, used with
conditional deallocation 98
conditional deallocation that includes data 106
entering receive state conditionally 241
entering receive state conditionally and including data 250
entering receive state unconditionally 274
entering receive state unconditionally and including data 262
flushing the buffer 482
sending data and flushing the buffer 456
sending data conditionally 415, 444
unconditional deallocation 129
unconditional deallocation that includes data 118
macroinstructions for sending new data, used with
conditional deallocation that includes data 106
entering receive state conditionally and including data 250
entering receive state unconditionally and including data 262
sending data 431
sending data and flushing the buffer 456
sending data conditionally 444
unconditional deallocation that includes data 118
sending expedited data on a full-duplex session 497
conversation deallocation
APPCCMD CONTROL=DEALLOC
macroinstruction syntax and operands 68, 129
APPCCMD CONTROL=DEALLOCQ
macroinstruction syntax and operands 136
use of macroinstruction 136
conversation identifier
keyword, used with
conditional deallocation 101
conditional deallocation that includes data 109
deallocation due to program errors 71
deallocation due to service errors 79
deallocation due to timer errors 86
deallocation due to user-specified criteria 93
entering receive state conditionally 244
entering receive state conditionally and including data 253
entering receive state unconditionally 276
entering receive state unconditionally and including data 265

conversation identifier *(continued)*
flushing the buffer 485
indicating the beginning of a synchronization exchange 538
obtain status on information available on a specified conversation 567
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available on a specific conversation 370
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
sending an error notification 472
sending data 435
sending data and flushing the buffer 459
sending data conditionally 418, 447
sending expedited data on a full-duplex session 501
specific-mode RECEIVEs 382
suspending a subsequent conversation 533
terminating a session 393
unconditional deallocation 132
unconditional deallocation that includes data 121
returned value, used with
allocation request, conditional, without wait 62
allocation request for contention-winner session 39
allocation request for specific session 26
allocation request that can be queued 15
any-mode RECEIVEs 349
immediate allocation request 50
receive normal information immediately available from any conversation 361
receiving an FMH-5 330
receiving expedited data in any-mode 285
conversation security
conversation level 634
CONVGRP qualify value
session assignment algorithm 19
CONVID
keyword, used with
conditional deallocation 101
conditional deallocation that includes data 109
deallocation due to program errors 71
deallocation due to service errors 79
deallocation due to timer errors 86
deallocation due to user-specified criteria 93
entering receive state conditionally 244
entering receive state conditionally and including data 253
entering receive state unconditionally 276
flushing the buffer 485
indicating the beginning of a synchronization exchange 538
obtain status on information available on a specified conversation 567
obtain status on information immediately available on a specified conversation 560
CONVID *(continued)*
keyword, used with *(continued)*
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available on a specific conversation 370
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
sending an error notification 472
sending data 435
sending data and flushing the buffer 459
sending data conditionally 418, 447
sending expedited data on a full-duplex session 501
specific-mode RECEIVEs 382
suspending a subsequent conversation 533
terminating a session 393
unconditional deallocation 132
unconditional deallocation that includes data 121
used with entering receive state unconditionally and including data 265
returned value, used with
allocation request, conditional, without wait 62
allocation request for specific session 26
allocation request that can be queued 15
any-mode RECEIVEs 349
immediate allocation request 50
receive normal information immediately available from any conversation 361
receiving an FMH-5 330
receiving expedited data from the specified conversation 307
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 299
receiving expedited data in any-mode 285
CONVSECP RPL field
returned value, used with
allocation request, conditional, without wait 62
allocation request for specific session 26
allocation requests for contention-winner session 39
allocation requests that can be queued 15
immediate allocation requests 50
CONWIN qualify value
session assignment algorithm 31
CONXMOD keyword, used with
allocation request, conditional, without wait 57
allocation request for contention-winner session 34
allocation request for specific session 23
allocation request that can be queued 10
any-mode RECEIVES 347
conditional deallocation 101
conditional deallocation that includes data 109, 121, 132
entering receive state conditionally 244
entering receive state conditionally and including data 253
entering receive state unconditionally 276

CONXMOD keyword, used with *(continued)*
entering receive state unconditionally and including data 265
flushing the buffer 485
immediate allocation request 46
ISTRPL6 578
receives normal information immediately available from any conversation 358
receives normal information immediately available on a specified conversation 370
receiving an FMH-5 328
receiving expedited data from the specified conversation 307
receiving expedited data immediately available in any-mode 291
receiving expedited data immediately from the specified conversation 299
receiving expedited data in any-mode 283
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
sending an error notification 472
sending data 435
sending data and flushing the buffer 459
sending data conditionally 418, 447
sending expedited data on full-duplex session 501
specific-mode RECEIVEs 382
cross-memory mode, restrictions 66
CRYPT
keyword, used with
conditional deallocation that includes data 110
entering receive state 254
entering receive state unconditionally 266
sending data 435
sending data and flushing the buffer 460
sending data conditionally 448
unconditional deallocation that includes data 122
CRYPTLVL RPL field
returned value, used with
allocation request, conditional, without wait 62
allocation request for contention-winner session 39
allocation request for specific session 26
allocation request that can be queued 15
immediate allocation request 50
receiving an FMH-5 331

D

DACTSESS qualify value
used with terminating a session 160
data, receiving
APPCCMD macroinstructions used 343, 378
data, sending
APPCCMD macroinstructions used
syntax and operands 415, 490
buffer list (OPTCD=BUFFLST) considerations
conditional deallocation that includes data 110
entering receive state conditionally and including data 255
sending data 112, 436
data, sending *(continued)*
buffer list (OPTCD=BUFFLST) considerations *(continued)*
sending data and flushing the buffer 255, 267, 461
sending data conditionally 449
sending expedited data on a full-duplex session 502
unconditional deallocation that includes data 124
flushing the buffer, used with
conditional deallocation 98
conditional deallocation that includes data 106
entering receive state conditionally 241
entering receive state conditionally and including data 250
entering receive state unconditionally 274
entering receive state unconditionally and including data 262
flushing the buffer 482
sending data and flushing the buffer 456
sending data conditionally 415, 444
unconditional deallocation 129
unconditional deallocation that includes data 118
macroinstructions for sending new data, used with
conditional deallocation that includes data 106
entering receive state conditionally and including data 250
entering receive state unconditionally and including data 262
sending data 431
sending data and flushing the buffer 456
sending data conditionally 444
unconditional deallocation that includes data 118
sending expedited data on a full-duplex session 497
DATA qualify value 431, 497
data truncation
sending an error notification macroinstruction 468
DATACON qualify value
DEALLOC CONTROL value 106
PREPRCV CONTROL value 250
SEND CONTROL value 444
DATAFLU qualify value
DEALLOC CONTROL value 118
PREPRCV CONTROL value 262
SEND CONTROL value 456
deactivating sessions
used with terminating a session macroinstruction 160
deactivation type code (DEACTYP)
keyword, used with
terminating session due to error 404
terminating session due to protocol error or cleanup deactivation 398
DEACTYP (deactivation type code)
keyword, used with
terminating session due to error 404
terminating session due to protocol error or cleanup deactivation 398

deallocating a conversation
APPCCMD CONTROL=DEALLOC
macroinstruction syntax and operands 68, 90
APPCCMD CONTROL=DEALLOCQ
macroinstruction syntax and operands 136
use of macroinstruction 136
deallocating conversations
macroinstruction forms of
CONTROL=DEALLOC 68, 129
deallocation, abnormal
LU Services errors (ABNDSERV) 76, 136
reference material (macroinstruction syntax and operands) 68, 90
timing errors (ABNDTIME) 83, 136
transaction program errors (ABNDPROG) 68, 136
user-defined errors (ABNDUSER) 90, 136
DEFINE qualify value
syntax and operands 165
defining LU-mode table values 165
design considerations for LU 6.2
restoring a mode 179
DISPLAY qualify value
syntax and operands 172
displaying LU-mode data 172
DSECT
ISTDBIND 625
ISTFM5 635
ISTRPL6X 637
ISTSLCNS 645
ISTSLD 646
ISTSREST 648
ISTSTATD 648
ISTUSFBC 649

E

ECB (event control block)
keyword, used with
accepting a session 147
allocation request, conditional, without wait 58
allocation request for contention-winner session 35
allocation request for specific session 23
allocation request that can be queued 11
any-mode RECEIVEs 347
CNOS requests 153
conditional deallocation 101
conditional deallocation that includes data 110
deallocation due to program errors 71
deallocation due to service errors 79
deallocation due to timer errors 86
deallocation due to user-specified criteria 93
defining LU-mode table values 168
displaying LU-mode table values 174
entering receive state conditionally 244
entering receive state conditionally and including data 254
entering receive state unconditionally 277
entering receive state unconditionally and including data 266
flushing the buffer 485
ECB (event control block) *(continued)*
keyword, used with *(continued)*
immediate allocation request 46
indicating the beginning of a synchronization exchange 538
indicating the end of a synchronization exchange 543
obtain status on information available on a specified conversation 567
obtain status on information from any active conversation 548
obtain status on information immediately available from any active conversation 554
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available from any conversation 359
receive normal information immediately available on a specific conversation 370
receiving an FMH-5 328
receiving expedited data from the specified conversation 308
receiving expedited data immediately available in any-mode 291
receiving expedited data immediately from the specified conversation 300
receiving expedited data in any-mode 283
rejecting a session 162
releasing a suspended session 528
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
restoring a mode 181
sending an error notification 472
sending data 435
sending data and flushing the buffer 460
sending data conditionally 418, 448
sending expedited data on a full-duplex session 501
specific-mode RECEIVEs 382
suspending a subsequent conversation 533
terminating a session 393
terminating a session due to errors 404
terminating a session due to protocol errors or cleanup deactivation 399
unconditional deallocation 132
unconditional deallocation that includes data 122
entering RECEIVE state
PREPRCV macroinstructions (syntax and operands) 241, 274
entering SEND state
requesting permission to enter send state macroinstruction 490
ERROR qualify value 468
event control block (ECB)
keyword, used with
accepting a session 147
allocation request, conditional, without wait 58

event control block (ECB) *(continued)*
allocation request for contention-winner session 35
allocation request for specific session 23
allocation request that can be queued 11
any-mode RECEIVEs 347
CNOS requests 153
conditional deallocation 101
conditional deallocation that includes data 110
deallocation due to program errors 71
deallocation due to service errors 79
deallocation due to timer errors 86
deallocation due to user-specified criteria 93
defining LU-mode table values 168
displaying LU-mode table values 174
entering receive state conditionally 244
entering receive state conditionally and including data 254
entering receive state unconditionally 277
entering receive state unconditionally and including data 266
flushing the buffer 485
immediate allocation request 46
indicating the beginning of a synchronization exchange 538
indicating the end of a synchronization exchange 543
obtain status on information available on a specified conversation 567
obtain status on information from any active conversation 548
obtain status on information immediately available from any active conversation 554
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available from any conversation 359
receive normal information immediately available on a specific conversation 370
receiving an FMH-5 328
receiving expedited data from the specified conversation 308
receiving expedited data immediately available in any-mode 291
receiving expedited data immediately from the specified conversation 300
receiving expedited data in any-mode 283
rejecting a session 162
releasing a suspended session 528
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
restoring a mode 181
sending an error notification 472
sending data 435
sending data and flushing the buffer 460
sending data conditionally 418, 448
sending expedited data on a full-duplex session 501

event control block (ECB) *(continued)*
keyword, used with *(continued)*
specific-mode RECEIVEs 382
suspending a subsequent conversation 533
terminating a session 393
terminating a session due to errors 404
terminating a session due to protocol errors or cleanup deactivation 399
unconditional deallocation 132
unconditional deallocation that includes data 122

EXIT keyword, used with
accepting a session 147
allocation request, conditional, without wait 58
allocation request for contention-winner session 35
allocation request for specific session 23
allocation request that can be queued 11
any-mode RECEIVEs 348
CNOS requests 153
conditional deallocation 101
conditional deallocation that includes data 110
deallocation due to program errors 71
deallocation due to service errors 79
deallocation due to timer errors 86
deallocation due to user-specified criteria 93
defining LU-mode table values 168
displaying LU-mode table values 175
entering receive state conditionally 244
entering receive state conditionally and including data 254
entering receive state unconditionally 277
entering receive state unconditionally and including data 266
flushing the buffer 486
immediate allocation request 47
indicating the beginning of a synchronization exchange 539
indicating the end of a synchronization exchange 543
queued deallocation due to program, service, timer, or user errors 284
receive normal information immediately available from any conversation 359
receive normal information immediately available on a specific conversation 371
receiving an FMH-5 329
receiving expedited data from the specified conversation 308
receiving expedited data immediately available in any-mode 292
rejecting a session 162
releasing a suspended session 528
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
restoring a mode 182
sending an error notification 472
sending data 436
sending data and flushing the buffer 460
sending data conditionally 418, 448
specific-mode RECEIVEs 383
suspending a subsequent conversation 534

EXIT keyword, used with *(continued)*
terminating a session 393
terminating a session due to errors 405
terminating a session due to protocol errors or cleanup deactivation 399
unconditional deallocation 132
unconditional deallocation that includes data 122
EXPDLEN RPL field
returned value, used with
any-mode RECEIVEs 349
conditional deallocation 102
conditional deallocation that includes data 112
deallocation due to program errors 72
deallocation due to service errors 80
deallocation due to timer errors 87
deallocation due to user-specified criteria 95
entering receive state conditionally and including data 257
entering receive state unconditionally and including data 268
receive normal information immediately available from any conversation 361
receive normal information immediately available on a specific conversation 373
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 301
receiving expedited data in any-mode 285
sending an error notification 475
sending data 438
sending data and flushing the buffer 463
sending data conditionally 419, 450
sending expedited data on full-duplex session 503
specific-mode RECEIVEs 385
unconditional deallocation 133
unconditional deallocation that includes data 124
EXPDRCV RPL field
returned value, used with
any-mode RECEIVEs 350
conditional deallocation 102
conditional deallocation that includes data 112
deallocation due to program errors 73
deallocation due to service errors 80
deallocation due to timer errors 87
deallocation due to user-specified criteria 95
entering receive state conditionally and including data 257
entering receive state unconditionally and including data 268
receive normal information immediately available from any conversation 361
receive normal information immediately available on a specific conversation 373
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 293
EXPDRCV RPL field *(continued)*
returned value, used with *(continued)*
receiving expedited data immediately from the specified conversation 301
receiving expedited data in any-mode 285
sending an error notification 475
sending data 438
sending data and flushing the buffer 463
sending data conditionally 419, 451
sending expedited data on full-duplex session 503
specific-mode RECEIVEs 385
unconditional deallocation 133
unconditional deallocation that includes data 124

F

FDB2 RPL field
returned value, used with
accepting a session 148
allocation request, conditional, without wait 62
allocation request for contention-winner session 39
allocation request for specific session 27
allocation request that can be queued 15
any-mode RECEIVEs 350
CNOS requests 156
conditional deallocation 102
conditional deallocation that includes data 112
deallocation due to program errors 73
deallocation due to service errors 80
deallocation due to timer errors 87
deallocation due to user-specified criteria 95
defining LU-mode table values 170
displaying LU-mode table values 177
entering receive state conditionally 246
entering receive state conditionally and including data 257
entering receive state unconditionally 278
entering receive state unconditionally and including data 268
flushing the buffer 487
immediate allocation request 50
indicating the beginning of a synchronization exchange 540
indicating the end of a synchronization exchange 544
obtain status on information available on a specified conversation 569
obtain status on information from any active conversation 549
obtain status on information immediately available on a specified conversation 562
queued deallocation due to program, service, timer, or user errors 141
receive normal information immediately available from any conversation 361
receive normal information immediately available on a specific conversation 373
receiving an FMH-5 331

FDB2 RPL field *(continued)*
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 301
receiving expedited data in any-mode 285
rejecting a session 163
releasing a suspended session 529
replying to a confirmation request 428
requesting permission to enter send state 495
resetting the continuation mode 413
restoring a mode 183
sending an error notification 475
sending data 438
sending data and flushing the buffer 463
sending data conditionally 420, 451
sending expedited data on full-duplex session 503
specific-mode RECEIVEs 385
suspending a subsequent conversation 535
terminating a session 394
terminating a session due to errors 406
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 133
unconditional deallocation that includes data 124
feedback code data structure—ISTUSFBC DSECT 649
feedback information
request-to-send indicator 490
FILL keyword
receive normal information immediately available on a specific conversation 371
used with ISTRPL6 578
used with specific-mode RECEIVEs 383
FLUSH qualify value
DEALLOC CONTROL value 129
PREPRCV CONTROL value 274
SEND CONTROL value 482
flushing the send buffer
flushing the buffer macroinstruction 482
used with entering receive state unconditionally macroinstruction 274
used with unconditional deallocation macroinstruction 129
FMH-5
DSECT 635
FMH5LEN RPL field
returned value, used with
allocation request, conditional, without wait 62
allocation request for contention-winner session 39
allocation request for specific session 27
allocation request that can be queued 15
any-mode RECEIVEs 350
conditional deallocation 102
conditional deallocation that includes data 113
deallocation due to program errors 73
deallocation due to service errors 80
FMH5LEN RPL field *(continued)*
returned value, used with *(continued)*
deallocation due to timer errors 87
deallocation due to user-specified criteria 95
entering receive state conditionally 246
entering receive state conditionally and including data 257
entering receive state unconditionally 278
entering receive state unconditionally and including data 268
flushing the buffer 487
immediate allocation request 50
obtain status on information available on a specified conversation 569
obtain status on information from any active conversation 549
obtain status on information immediately available from any active conversation 555
obtain status on information immediately available on a specified conversation 562
queued deallocation due to program, service, timer, or user errors 141
receive normal information immediately available from any conversation 361
receive normal information immediately available on a specific conversation 373
receiving an FMH-5 331
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 301
receiving expedited data in any-mode 285
replying to a confirmation request 428
requesting permission to enter send state 495
resetting the continuation mode 413
sending an error notification 475
sending data 438
sending data and flushing the buffer 463
sending data conditionally 420, 451
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 385
terminating a session 394
terminating a session due to errors 406
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 134
unconditional deallocation that includes data 125
FMH5RCV RPL field
returned value, used with
allocation request, conditional, without wait 62
allocation request for contention-winner session 39
allocation request for specific session 27
allocation request that can be queued 15
any-mode RECEIVEs 350
conditional deallocation 102
conditional deallocation that includes data 113
deallocation due to program errors 73

FMH5RCV RPL field *(continued)*
deallocation due to service errors 80
deallocation due to timer errors 88
deallocation due to user-specified criteria 95
entering receive state conditionally 246
entering receive state conditionally and including data 257
entering receive state unconditionally 278
entering receive state unconditionally and including data 268
flushing the buffer 487
immediate allocation request 50
obtain status on information available on a specified conversation 569
obtain status on information from any active conversation 549
obtain status on information immediately available from any active conversation 555
obtain status on information immediately available on a specified conversation 562
queued deallocation due to program, service, timer, or user errors 141
receive normal information immediately available from any conversation 361
receive normal information immediately available on a specific conversation 373
receiving an FMH-5 331
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 301
receiving expedited data in any-mode 285
replying to a confirmation request 428
requesting permission to enter send state 495
resetting the continuation mode 413
sending an error notification 475
sending data 438
sending data and flushing the buffer 463
sending data conditionally 420, 451
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 385
terminating a session 394
terminating a session due to errors 406
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 134
unconditional deallocation that includes data 125

G

global variables
declaring and setting with ISTGAPPC macroinstruction 572
table of 572

I

identifier, conversation
keyword, used with
conditional deallocation 101
conditional deallocation that includes data 109
deallocation due to program errors 71
deallocation due to service errors 79
deallocation due to timer errors 86
deallocation due to user-specified criteria 93
entering receive state conditionally 244
entering receive state conditionally and including data 253
entering receive state unconditionally 276
entering receive state unconditionally and including data 265
flushing the buffer 485
indicating the beginning of a synchronization exchange 538
obtain status on information available on a specified conversation 567
obtain status on information immediately available on a specified conversation 560
queued deallocation due to program, service, timer, or user errors 139
receive normal information immediately available on a specific conversation 370
replying to a confirmation request 427
requesting permission to enter send state 493
resetting the continuation mode 411
sending an error notification 472
sending data 435
sending data and flushing the buffer 459
sending data conditionally 418, 447
sending expedited data on a full-duplex session 501
specific-mode RECEIVEs 382
suspending a subsequent conversation 533
terminating a session 393
unconditional deallocation 132
unconditional deallocation that includes data 121
returned value, used with
allocation request, conditional, without wait 62
allocation request for contention-winner session 39
allocation request for specific session 26
allocation request that can be queued 15
any-mode RECEIVEs 349
immediate allocation request 50
receive normal information immediately available from any conversation 361
receiving an FMH-5 330
receiving expedited data in any-mode 285
IMMED qualify value
session assignment algorithm 43
use of 43
ISTDBIND DSECT 625
ISTFM5 DSECT 635
ISTGAPPC macroinstruction 572
ISTRPL6 macroinstruction 575
ISTRPL6X DSECT 637
ISTSLCNS DSECT 645

ISTSLD DSECT 646
ISTSREST DSECT 648
ISTUSFBC DSECT 649

L

limited resource
  session 635
LIST
  keyword, used with
    ISTRPL6 579
    restoring a mode 182
LOCKS keyword, used with
  entering receive state conditionally 244
  entering receive state conditionally and including data 254
  ISTRPL6 579
LOGMODE
  keyword, used with
    allocation request, conditional, without wait 58
    allocation request for contention-winner session 35
    allocation request that can be queued 11
    CNOS requests 153
    defining LU-mode table values 168
    displaying LU-mode table values 175
    immediate allocation request 47
    ISTRPL6 579
    restoring a mode 182
  returned value, used with
    allocation request for specific session 27
    receiving an FMH-5 331
LOGRCV RPL field
  returned value, used with
    any-mode RECEIVEs 350
    conditional deallocation 103
    conditional deallocation that includes data 113
    entering receive state conditionally 246
    entering receive state conditionally and including data 257
    entering receive state unconditionally and including data 269
    receive normal information immediately available from any conversation 361
    receive normal information immediately available on a specific conversation 373
    sending an error notification 475
    sending data 439
    sending data and flushing the buffer 463
    sending data conditionally 420, 451
    specific-mode RECEIVEs 385
LU 6
  restoring a mode 179
LU-mode table
  changing information in table 165
  querying information in table 172
LUAFFIN keyword, used with
  allocation request, conditional, without wait 58, 579
  allocation request for a contention-winner session 36
  allocation request for a specific session 23
  allocation request that can be queued 12
LUAFFIN keyword, used with *(continued)*
  CNOS requests 153
  reserving a contention-winner session without establishing conversation 212
  reserving a conversation session without establishing conversation 234
  reserving a session without establishing conversation 190
    with a specified conversation group 201
LUNAME
  keyword, used with
    allocation request, conditional, without wait 59
    allocation request for contention-winner session 36
    allocation request that can be queued 12
    CNOS requests 154
    defining LU-mode table values 168
    displaying LU-mode table values 175
    immediate allocation request 47
    ISTRPL6 580
    restoring a mode 182
  returned value, used with
    allocation request for specific session 28
    receiving an FMH-5 331

M

macroinstruction
  operands 1
  syntax 1
mode
  cross-memory, restrictions 66
  restoring 179

N

NAMEUSE
  keyword, used with
    allocation request, conditional, without wait 59
    allocation request for contention-winner session 36
    allocation request for specific session 24
    allocation request that can be queued 12
    CNOS requests 154
NETID
  keyword, used with
    allocation request, conditional, without wait 59
    allocation request for contention-winner session 36
    allocation request that can be queued 12
    CNOS requests 154
    defining LU-mode tables 169
    displaying LU-mode table values 175
    immediate allocation request 47
    ISTRPL6 580
    restoring a mode 182
  returned value, used with
    allocation request for specific session 28
    receiving an FMH-5 331

O

operand format 2
operand notation 2
OPTCD keyword, used with
  accepting a session 147
  allocation request, conditional, without wait 60
  allocation request for contention-winner session 37
  allocation request for specific session 24
  allocation request that can be queued 13
  any-mode RECEIVEs 348
  CNOS requests 155
  conditional deallocation 101
  conditional deallocation that includes data 110
  deallocation due to program errors 71
  deallocation due to service errors 79
  deallocation due to timer errors 86
  deallocation due to user-specified criteria 93
  defining LU-mode table values 169
  displaying LU-mode table values 175
  entering receive state conditionally 245
  entering receive state conditionally and including data 255
  entering receive state unconditionally 277
  entering receive state unconditionally and including data 266
  flushing the buffer 486
  immediate allocation request 48
  indicating the beginning of a synchronization exchange 539
  indicating the end of a synchronization exchange 543
  obtain status on information from any active conversation 549
  obtain status on information immediately available from any active conversation 555
    obtain status on information available on a specified conversation 568
    obtain status on information immediately available on a specified conversation 561
  queued deallocation due to program, service, timer, or user errors 140
  receive normal information immediately available from any conversation 359
  receive normal information immediately available on a specific conversation 371
  receiving an FMH-5 329
  receiving expedited data from the specified conversation 308
  receiving expedited data immediately available in any-mode 292
  receiving expedited data immediately from the specified conversation 300
  receiving expedited data in any-mode 284
  rejecting a session 162
  releasing a suspended session 528
  replying to a confirmation request 427
  requesting permission to enter send state 494
  resetting the continuation mode 412
  restoring a mode 183
  sending an error notification 472
  sending data 436
OPTCD keyword, used with *(continued)*
  sending data and flushing the buffer 460
  sending data conditionally 419, 448
  sending expedited data on a full-duplex session 502
  specific-mode RECEIVEs 383
  suspending a subsequent conversation 534
  terminating a session 393
  terminating a session due to errors 405
  terminating a session due to protocol errors or cleanup deactivation 399
  unconditional deallocation 132
  unconditional deallocation that includes data 122

P

parameters, session
  DSECT 625
  relationship to
    RESTORE 179
PIP (program initialization parameters) data
  DSECT 635
preparing to receive data 241, 274
PREPRCV macroinstructions (syntax and operands) . 241, 274
program initialization parameters (PIP) data
  DSECT 635
PRSISTVP RPL field
  returned value, used with
    allocation request, conditional, without wait 63
    allocation request for contention-winner session 40
    allocation request for specific session 28
    allocation request that can be queued 16
    CNOS request 156
    immediate allocation request 51
PSERVIC
  profile 1 628
  profile 2 631
  profile 3 632
  profile 4 632
  profile 6 634

Q

QUALIFY keyword
  ABNDPROG value 68, 136
  ABNDSERV value 76, 136
  ABNDTIME value 83, 136
  ABNDUSER value 90, 136
  ACTSESS value 144
  ALLOCD value 7
  ANY value
    RCVEXPD CONTROL value 280, 288, 296, 304
    RECEIVE CONTROL value 343, 355
  CNOS value 150
  CONFIRM value
    CONFRMD QUALIFY value 424
    DEALLOC CONTROL value 98
    PREPRCV CONTROL value 241
    SEND CONTROL value 415

QUALIFY keyword *(continued)*
CONV value 390
CONVGRP value 19, 396
CONWIN value 31
DACTSESS value 160
DATA value
SEND CONTROL value 431
SENDEXPD CONTROL value 497
DATACON value
DEALLOC CONTROL value 106
PREPRCV CONTROL value 250
SEND CONTROL value 444
DATAFLU value
DEALLOC CONTROL value 118
PREPRCV CONTROL value 262
SEND CONTROL value 456
DEFINE value 165
DISPLAY value 172
ERROR value 468
FLUSH value
DEALLOC CONTROL value 129
PREPRCV CONTROL value 274
SEND CONTROL value 482
IMMED value 43, 220
ISPEC value 366
RESTORE value 179
RESUME value 526
RQSEND value 490
SESSION value 402
SPEC value 378
SUSPEND value 531
SYNCBEG value 536
SYNCEND value 541
used with ISTRPL6 580
WHENFREE value 54

R

RCPRI
returned value, used with
accepting a session 148
allocation request, conditional, without wait 63
allocation request for contention-winner session 40
allocation request for specific session 28
allocation request that can be queued 16
any-mode RECEIVEs 351
CNOS requests 157
conditional deallocation 103
conditional deallocation that includes data 114
deallocation due to program errors 73
deallocation due to service errors 81
deallocation due to timer errors 88
deallocation due to user-specified criteria 95
defining LU-mode table values 170
displaying LU-mode table values 177
entering receive state conditionally 247
entering receive state conditionally and including data 258
entering receive state unconditionally 278

RCPRI *(continued)*
returned value, used with *(continued)*
entering receive state unconditionally and including data 269
flushing the buffer 487
immediate allocation request 51
indicating the beginning of a synchronization exchange 540
indicating the end of a synchronization exchange 544
obtain status on information available on a specified conversation 570
obtain status on information from any active conversation 550
obtain status on information immediately available from any active conversation 556
obtain status on information immediately available on a specified conversation 563
queued deallocation due to program, service, timer, or user errors 142
receive normal information immediately available from any conversation 362
receive normal information immediately available on a specific conversation 374
receiving an FMH-5 331
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 285
rejecting a session 163
releasing a suspended session 529
replying to confirmation request 429
requesting permission to enter send state 495
resetting the continuation mode 413
restoring a mode 184
sending an error notification 476
sending data 439
sending data and flushing the buffer 464
sending data conditionally 421, 452
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 386
suspending a subsequent conversation 535
terminating a session 394
terminating a session due to errors 406
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 134
unconditional deallocation that includes data 125

RCSEC
returned value, used with
accepting a session 148
allocation request, conditional, without wait 63
allocation request for specific session 28
allocation request that can be queued 16
allocation requests for contention-winner session 40
any-mode RECEIVEs 351

RCSEC *(continued)*
CNOS requests 157
conditional deallocation 104
conditional deallocation that includes data 114
deallocation due to program errors 73
deallocation due to service errors 81
deallocation due to timer errors 88
deallocation due to user-specified criteria 95
defining LU-mode table values 170
displaying LU-mode table values 177
entering receive state conditionally 247
entering receive state conditionally and including data 258
entering receive state unconditionally 278
entering receive state unconditionally and including data 270
flushing the buffer 487
immediate allocation request 51
indicating the beginning of a synchronization exchange 540
indicating the end of a synchronization exchange 544
obtain status on information available on a specified conversation 570
obtain status on information from any active conversation 550
obtain status on information immediately available from any active conversation 556
obtain status on information immediately available on a specified conversation 563
queued deallocation due to program, service, timer, or user errors 142
receive normal information immediately available from any conversation 362
receive normal information immediately available on a specific conversation 374
receiving an FMH-5 332
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 293
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 285
rejecting a session 163
releasing a suspended session 529
replying to a confirmation request 429
requesting permission to enter send state 495
resetting the continuation mode 413
restoring a mode 184
sending an error notification 476
sending data 439
sending data and flushing the buffer 464
sending data conditionally 421, 452
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 386
suspending a subsequent conversation 535
terminating a session 394
terminating a session due to errors 407

RCSEC *(continued)*
returned value, used with *(continued)*
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 134
unconditional deallocation that includes data 125
RCVFMH5 control value 325
receive-any mode
used with any-mode RECEIVEs macroinstruction 343
receive-specific mode
any-mode RECEIVEs macroinstruction, used with 378
RECEIVE state, entering
PREPRCV macroinstructions (syntax and operands) 241, 274
receiving data
APPCCMD macroinstructions used 343, 378
RECLEN
keyword, used with
accepting a session 148
allocation request, conditional, without wait 60
allocation request for contention-winner session 37
allocation request for specific session 25
allocation request that can be queued 13
CNOS requests 155
conditional deallocation that includes data 112
deallocation due to program errors 72
deallocation due to service errors 79
deallocation due to timer errors 87
deallocation due to user-specified criteria 94
defining LU-mode table values 169
entering receive state conditionally and including data 256
entering receive state unconditionally and including data 268
immediate allocation request 48
queued deallocation due to program, service, timer, or user errors 140
sending an error notification 473
sending data 437
sending data and flushing the buffer 462
sending data conditionally 450
sending expedited data on a full-duplex session 503
unconditional deallocation that includes data 124
returned value, used with
any-mode RECEIVEs 351
displaying LU-mode table values 177
obtain status on information available on a specified conversation 570
obtain status on information from any active conversation 550
obtain status on information immediately available from any active conversation 556
obtain status on information immediately available on a specified conversation 563
receive normal information immediately available from any conversation 362

RECLEN *(continued)*
receive normal information immediately available on a specific conversation 374
receiving an FMH-5 332
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 294
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 286
restoring a mode 184
specific-mode RECEIVEs 386
reducing session limits 150
register usage
summary 675
request parameter list (RPL)
keyword, used with
accepting a session 148
allocation request, conditional, without wait 60
allocation request for contention-winner session 37
allocation request for specific session 25
allocation request that can be queued 13
any-mode RECEIVEs 349
CNOS requests 155
conditional deallocation 102
conditional deallocation that includes data 112
CONTROL=CHECK 67
deallocation due to program errors 72
deallocation due to service errors 80
deallocation due to timer errors 87
deallocation due to user-specified criteria 94
defining LU-mode table values 169
displaying LU-mode table values 176
entering receive state conditionally 245
entering receive state conditionally and including data 256
entering receive state unconditionally 278
entering receive state unconditionally and including data 268
flushing the buffer 486
immediate allocation request 48
indicating the beginning of a synchronization exchange 539
indicating the end of a synchronization exchange 544
obtain status on information available on a specified conversation 569
obtain status on information from any active conversation 549
obtain status on information immediately available from any active conversation 555
obtain status on information immediately available on a specified conversation 562
queued deallocation due to program, service, timer, or user errors 140
receive normal information immediately available from any conversation 360
receive normal information immediately available on a specific conversation 372
request parameter list (RPL) *(continued)*
keyword, used with *(continued)*
receiving an FMH-5 329
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 292
receiving expedited data immediately from the specified conversation 300
receiving expedited data in any-mode 284
rejecting a session 163
releasing a suspended session 529
replying to a confirmation request 428
requesting permission to enter send state 494
resetting the continuation mode 412
restoring a mode 183
sending an error notification 473
sending data 437
sending data and flushing the buffer 462
sending data conditionally 419, 450
sending expedited data on a full-duplex session 503
specific-mode RECEIVEs 384
suspending a subsequent conversation 534
terminating a session 394
terminating a session due to errors 405
terminating a session due to protocol errors or cleanup deactivation 399
unconditional deallocation 133
unconditional deallocation that includes data 124
RESETRCV control value 408
responding negatively to session requests
used with terminating a session 160
responding positively to session requests 144
RESTORE macroinstruction 179
RESTORE qualify value 179
restoring modes and sessions 179
return codes
listed and described 585
RCPRI, RCSEC combinations 585
RTNCD, FDB2 information 623
returned information
request-to-send indicator 490
RPL (request parameter list)
keyword, used with
accepting a session 148
allocation request, conditional, without wait 60
allocation request for contention-winner session 37
allocation request for specific session 25
allocation request that can be queued 13
any-mode RECEIVEs 349
CNOS requests 155
conditional deallocation 102
conditional deallocation that includes data 112
CONTROL=CHECK 67
deallocation due to program errors 72
deallocation due to service errors 80
deallocation due to timer errors 87
deallocation due to user-specified criteria 94
defining LU-mode table values 169

RPL (request parameter list) *(continued)*
displaying LU-mode table values 176
entering receive state conditionally 245
entering receive state conditionally and including data 256
entering receive state unconditionally 278
entering receive state unconditionally and including data 268
flushing the buffer 486
immediate allocation request 48
indicating the beginning of a synchronization exchange 539
indicating the end of a synchronization exchange 544
obtain status on information available on a specified conversation 569
obtain status on information from any active conversation 549
obtain status on information immediately available from any active conversation 555
obtain status on information immediately available on a specified conversation 562
queued deallocation due to program, service, timer, or user errors 140
receive normal information immediately available from any conversation 360
receive normal information immediately available on a specific conversation 372
receiving an FMH-5 329
receiving expedited data from the specified conversation 309
receiving expedited data immediately available in any-mode 292
receiving expedited data immediately from the specified conversation 300
receiving expedited data in any-mode 284
rejecting a session 163
releasing a suspended session 529
replying to a confirmation request 428
requesting permission to enter send state 494
resetting the continuation mode 412
restoring a mode 183
sending an error notification 473
sending data 437
sending data and flushing the buffer 462
sending data conditionally 419, 450
sending expedited data on a full-duplex session 503
specific-mode RECEIVEs 384
suspending a subsequent conversation 534
terminating a session 394
terminating a session due to errors 405
terminating a session due to protocol errors or cleanup deactivation 399
unconditional deallocation 133
unconditional deallocation that includes data 124

RPL extension control block 637

RPL extension layout, figure 643

RPL extension—ISTRPL6X DSECT 637

RTNCD field
returned value, used with
accepting a session 148
allocation request, conditional, without wait 63
allocation request for contention-winner session 40
allocation request for specific session 28
allocation request that can be queued 16
any-mode RECEIVEs 351
CNOS requests 157
conditional deallocation 104
conditional deallocation that includes data 114
deallocation due to program errors 73
deallocation due to service errors 81
deallocation due to timer errors 88
deallocation due to user-specified criteria 96
defining LU-mode table values 170
displaying LU-mode table values 177
entering receive state conditionally 247
entering receive state conditionally and including data 258
entering receive state unconditionally 279
entering receive state unconditionally and including data 270
flushing the buffer 488
immediate allocation request 51
indicating the beginning of a synchronization exchange 540
indicating the end of a synchronization exchange 544
obtain status on information available on a specified conversation 570
obtain status on information from any active conversation 550
obtain status on information immediately available from any active conversation 556
obtain status on information immediately available on a specified conversation 563
queued deallocation due to program, service, timer, or user errors 142
receive normal information immediately available from any conversation 362
receive normal information immediately available on a specific conversation 374
receiving an FMH-5 332
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 294
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 286
rejecting a session 163
releasing a suspended session 530
replying to a confirmation request 429
requesting permission to enter send state 495
resetting the continuation mode 413
restoring a mode 184
sending an error notification 476
sending data 440
sending data and flushing the buffer 464

RTNCD field *(continued)*
sending data conditionally 421, 452
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 386
suspending a subsequent conversation 535
terminating a session 395
terminating a session due to errors 407
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 134
unconditional deallocation that includes data 125
RTSRTRN keyword, used with
allocation request, conditional, without wait 60
allocation request for contention-winner session 37
allocation request for specific session 25
allocation request that can be queued 13
immediate allocation request 48
ISTRPL6 582
receiving an FMH-5 329

S

security
conversation level 634
sending data
APPCCMD macroinstructions used
syntax and operands 415, 490
buffer list (OPTCD=BUFFLST) considerations
conditional deallocation that includes data 110
entering receive state conditionally and including data 255
sending data 112, 436
sending data and flushing the buffer 255, 267, 461
sending data conditionally 449
sending expedited data on a full-duplex session 502
unconditional deallocation that includes data 124
flushing the buffer, used with
conditional deallocation 98
conditional deallocation that includes data 106
entering receive state conditionally 241
entering receive state conditionally and including data 250
entering receive state unconditionally 274
entering receive state unconditionally and including data 262
flushing the buffer 482
sending data and flushing the buffer 456
sending data conditionally 415, 444
unconditional deallocation 129
unconditional deallocation that includes data 118
macroinstructions for sending new data, used with
conditional deallocation that includes data 106
entering receive state conditionally and including data 250
entering receive state unconditionally and including data 262
sending data 431
sending data and flushing the buffer 456
sending data *(continued)*
macroinstructions for sending new data, used with *(continued)*
sending data conditionally 444
unconditional deallocation that includes data 118
sending expedited data on a full-duplex session 497
SENSE
keyword, used with
accepting a session 148
deallocation due to user-specified criteria 94
ISTRPL6 583
sending an error notification 473
terminating a session 394
terminating a session due to errors 405
terminating a session due to protocol errors or cleanup deactivation 400
returned value, used with
allocation request, conditional, without wait 63
allocation request for contention-winner session 40
allocation request for specific session 28
allocation request that can be queued 16
any-mode RECEIVEs 351
CNOS request 157
conditional deallocation 104
conditional deallocation that includes data 114
entering receive state conditionally 248
entering receive state conditionally and including data 258
entering receive state unconditionally and including data 270
receive normal information immediately available from any conversation 363
receive normal information immediately available on a specific conversation 374
receiving an FMH-5 332
sending an error notification 476
sending data 440
sending data and flushing the buffer 465
sending data conditionally 421, 452
specific-mode RECEIVEs 386
unconditional deallocation that includes data 125
SESSID
keyword, used with
indicating the beginning of a synchronization exchange 539
indicating the end of a synchronization exchange 544
terminating a session 406
returned value, used with
allocation request, conditional, without wait 63
allocation request for contention-winner session 41
allocation request for specific session 28
allocation request that can be queued 17
immediate allocation request 51
receiving an FMH-5 332
releasing a suspended session 529
suspending a subsequent conversation 534

SESSIDL
keyword, used with
indicating the beginning of a synchronization exchange 539
indicating the end of a synchronization exchange 544
releasing a suspended session 529
terminating a session due to errors 406
returned value, used with
allocation request, conditional, without wait 64
allocation request for contention-winner session 41
allocation request for specific session 29
allocation request that can be queued 17
immediate allocation request 51
receiving an FMH-5 332
suspending a subsequent conversation 534
session activation, macroinstruction for 144
session deactivation
used with terminating a session macroinstruction 160
session limit
changing with CNOS requests 150
session limits, changing
with CNOS request macroinstruction 150
session parameters.
DSECT 625
relationship to
RESTORE 179
session requests, accepting
with a session macroinstruction 144
session requests, rejecting
used with terminating a session 160
SETSESS control value 526, 531, 536, 541
SIGDATA RPL extension field
returned value, used with
any-mode RECEIVEs 351
conditional deallocation that includes data 114
entering receive state conditionally and including data 259
entering receive state unconditionally and including data 270
receive normal information immediately available on a specific conversation 374
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 294
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 286
sending an error notification 476
sending data 440
sending data and flushing the buffer 465
sending data conditionally 421, 452
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 387
unconditional deallocation that includes data 125
Signal RU 490
SIGRCV RPL extension field
returned value, used with
any-mode RECEIVEs 352
conditional deallocation that includes data 115
entering receive state conditionally and including data 259
entering receive state unconditionally and including data 270
receive normal information immediately available from any conversation 363
receive normal information immediately available on a specific conversation 375
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 294
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 286
sending an error notification 477
sending data 440
sending data and flushing the buffer 465
sending data conditionally 421, 453
sending expedited data on full-duplex session 504
specific-mode RECEIVEs 387
unconditional deallocation that includes data 126
SLS RPL field
returned value, used with
allocation request, conditional, without wait 64
allocation request for contention-winner session 41
allocation request for specific session 29
allocation requests that can be queued 17
immediate allocation request 51
receiving an FMH-5 332
specific-mode RECEIVEs
any-mode RECEIVEs macroinstruction, used with 378
STSHBF
returned value, used with
conditional deallocation that includes data 115
deallocation due to program errors 73
deallocation due to service errors 81
deallocation due to timer errors 88
deallocation due to user-specified errors 96
entering receive state conditionally and including data 259
entering receive state unconditionally and including data 271
queued deallocation due to program, service, timer, or user errors 142
sending an error notification 477
sending data 441
sending data and flushing the buffer 465
sending data conditionally 453
unconditional deallocation that includes data 126
STSHDS
returned value, used with
conditional deallocation that includes data 115
deallocation due to service errors 81

STSHDS *(continued)*
deallocation due to timer errors 88
deallocation due to user-specified errors 96
entering receive state conditionally and including data 259
entering receive state unconditionally and including data 271
queued deallocation due to program, service, timer, or user errors 142
sending an error notification 477
sending data 441
sending data and flushing the buffer 465
sending data conditionally 453
unconditional deallocation that includes data 126
sync level 634

T

terminating session
terminating a session due to errors (syntax and operands) 402
truncating
sending an error notification macroinstruction 468
TYPE keyword, used with
ISTRPL6 583
sending an error notification 474

U

USERFLD
keyword, used with
accepting a session 148
allocation request, conditional, without wait 60
allocation request for contention-winner session 38
allocation request for specific session 25
allocation request that can be queued 14
CNOS request 155
defining LU-mode table values 170
displaying LU-mode table values 176
immediate allocation request 48
ISTRPL6 583
receiving an FMH-5 330
rejecting a session 163
restoring a mode 183
returned value, used with
accepting a session 149
any-mode RECEIVEs 352
CNOS request 157
conditional deallocation 104
conditional deallocation that includes data 115
deallocation due to program errors 74
deallocation due to service errors 81
deallocation due to timer errors 88
deallocation due to user-specified criteria 96
defining LU-mode table values 170
displaying LU-mode table values 177
entering receive state conditionally 248
entering receive state conditionally and including data 259
entering receive state unconditionally 279
USERFLD *(continued)*
returned value, used with *(continued)*
entering receive state unconditionally and including data 271
flushing the buffer 488
obtain status on information available on a specified conversation 570
obtain status on information immediately available on a specified conversation 563
queued deallocation due to program, service, timer, or user errors 142
receive normal information immediately available from any conversation 363
receive normal information immediately available on a specific conversation 375
receiving expedited data from the specified conversation 310
receiving expedited data immediately available in any-mode 294
receiving expedited data immediately from the specified conversation 302
receiving expedited data in any-mode 286
rejecting a session 163
replying to a confirmation request 429
requesting permission to enter send state 495
resetting the continuation mode 414
restoring a mode 184
sending an error notification 477
sending data 441
sending data and flushing the buffer 466
sending data conditionally 422, 453
sending expedited data on full-duplex session 505
specific-mode RECEIVEs 387
terminating a session 395
terminating a session due to protocol errors or cleanup deactivation 400
unconditional deallocation 134
unconditional deallocation that includes data 126

W

what-received RPL extension field
returned value, used with
any-mode RECEIVEs 352, 363
receive normal information immediately available from any conversation 363
receive normal information immediately available on a specific conversation 375
specific-mode RECEIVEs 387
WHENFREE qualify value
session assignment algorithm 54
winner, contention
requests for allocation 31

Readers' Comments — We'd Like to Hear from You

z/OS Communications Server
SNA Programmer's LU 6.2 Reference
Version 1 Release 2

Publication No. SC31-8810-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC31-8810-00

IBM®

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Software Reengineering
Department G7IA/ Bldg 503
Research Triangle Park, NC
27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

SC31-8810-00

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8810-00

Spine information:

z/OS Communications Server

z/OS V1R2.0 CS: SNA Programmer's LU 6.2 Reference

Version 1
Release 2