eNetwork 通E服务器f本 6.0
Windows NT f和 eNetwork 个人通E f本**
4.2 Windows 95 和 Windows NT f**

系统管m程 r h F

SC84-0693-00

eNetwork 通E服务器f本 6.0
Windows NT f和 eNetwork 个人通E f本**
4.2 Windows 95 和 Windows NT f**

系统管m程 r h F

SC84-0693-00

" b

在 9 C > 资 O 和 | 支 V D z 品 0， k 阅读 Z 619页 D 『 = < B. 注意 B 项』中 D 信息。

第二 f（1998 j 7 月）

> f > J C Z IBM eNetwork 通信服务器 Windows NT f， f > 6.0; v 人通信 Windows 95 和 Windows NT f， f > 4.2; 及 y P 后继发行 f 或修订 f， 直 = 在新 f > 中 m P y w 为止。

© Copyright International Business Machines Corporation 1989, 1997, 1998. All rights reserved.

? <

© Copyright IBM Corp. 1989, 1997, 1998

表

© Copyright IBM Corp. 1989, 1997, 1998

关 Z 本书

> i h v 如何 * 发 9 C IBM eNetwork 通信服务器 Windows NT f 以及 IBM eNetwork v 人通信 Windows 95 f 和 Windows NT f 。

IBM eNetwork 通信服务器 Windows NT f （在 > i 中 F 为*通信服务器*）G 一 v 通信服务平 (。 C 平 (为 k 主机和其 | 工作 > x 行通信 D Windows NT 工作 > a 供广泛 D 服务。通信服务器 C 户 I 从多种远 L ，通性选项中 x 行选择。

IBM eNetwork v 人通信 Windows 95 和 Windows NT f （在 > i 中 F 为 *v 人通信*）G 一 v 全功能仿 f 器。 } 主机终端仿 f 外， | 还 a 供下 P P C 功能:

- 文件传 d
- 动 , 配置
- 一 v 易 Z 9 C D 图形 g f
- 基 Z SNA M户机 & C L 序 D API
- 一 v 允许基 Z TCP/IP D & C L 序在基 Z SNA D 网 g O x 行通信 D API。

！管在大多 } i v 下， * 发 v 人通信和通信服务器 D L 序非 # 相 F ，因为 | G 都支 V 许多相同 D 动词， + 仍存在一些 n 异。 b 些 n 异通过图 j D 9 C 来 m > 。 N 阅 Z xii 页 D 『图 j 』 K b X 定细 Z 。在 > i 中， *L 序*指 v 人通信和通信服务器 = _ 。 1 v J C Z v 人通信 L 序或 v J C Z 通信服务器 L 序 1 ， + 9 C X 定 L 序 { 。

在 > i 中， Windows** 指 Windows 95 和 Windows NT**。在 > i 中， *工作 >* 指 y P 支 V D v 人计 c 机。 1 v 指一种 v 人计 c 机型号或 e 系 a 构 1 ，则 v C 类型为指定 D 。

本书的阅读对象

> i G 为那些要 9 C Z c Y 作员 h ）（NOF）API 消息来管理和 i 询 v 人通信或通信服务器 D Y 作， 和/或要 9 C ASCII 配置文件 D L 序员和 * 发 _ a 供 D 。

> i 也 I 为那些 ` 写网 g 管理 & C L 序 D * 发 _ 9 C， b 些 & C L 序 9 C I v 人通信和通信服务器 a 供 D 基 > 管理服务支 V 来 k 远 L （主机 9 c ）网 g 管理 & C L 序 x 行通信。

如何使 C 本书

> i 分为 = v ？分。 Z 1 页 D 『 Z 1 ？分 v 人通信和通信服务器 Z c Y 作员 h ）』 | 含下 P B Z :

- Z 5 页 D 『 Z 1 B i \ 』， h v > i D C 途。
- Z 9 页 D 『 Z 2 B > i 中动词 D E v 』， h v Z c Y 作员 h ） API a 构及其支 V D 动词。 C B E v 5 现动词 D 类 p 以及 v 人通信和通信服务器 a 供 D 其 { 信号。
- Z 19 页 D 『 Z 3 B Z c Y 作员 h) 入 Z c 』， h v 入 Z c 扩 9 。
- Z 4 = 12 B h v ？ v 动词 D o 法。 | 含存放 ？ v 动词信息 D a 构 D = 4 以及 ？一项 D h v ，后 z I 能返回 k D P m 。

© Copyright IBM Corp. 1989, 1997, 1998

Z 585页D『 Z 2？分 v 人通信和通信服务器管理服务 API』，｜含下P B Z:

- Z 587页D『 Z 13B 管理服务 API D i \ 』， h v管理服务 API。
- Z 591页D『 Z 14B 管理服务入Z c 』， h v管理服务动词D入Z c 。
- Z 603页D『 Z 15B 管理服务动词』， h v ？v动词D o法。｜含存放？v动词信息D a构D = 4以及？一项D h v，后z I能返回k D P m。

图标

在> i 中， 1需要B v X b信息1， + v现下P图 j :

此图 j m > I 0响v人通信或通信服务器D Y作，或任务完 I D注意B项或重要信息。

1信息v J C Z v人通信L序1 v现此图 j 。

1信息v J C Z通信服务器L序1 v现此图 j 。

本书中使C的约定

下P约定在{ v v人通信或通信服务器图 i b中9 C。 P v D某些约定 I 能未在> i 中9 C。

文本约定

粗体V	粗e字类型m > I在L序中或｜n a > 1 9 C D动词、函}和N }。 b些值x分大小写， y以& 1严q 4 ｜G在文>中v现D那样d人。
斜e字	斜e字类型m >下P: • 为其a供值D d ？。 • 窗Z X制D{ F，如P m、 4选r 、 d入字段、 4钮和K %选项。｜G在文>中v现M如｜G v现在窗Z中一样。 • i D j b 。 • 一v C作字母D字母，或一v C作%词D %词。}如: 1 4 = 一v *a* 1，确定｜在b里；会G一v *an*。
粗体 1 体V	粗e斜e字类型C Z ？w一v %词。
大写字e	大写字e m > I在L序中或｜n a > 1 9 C D # }、文件{、关键字和选项。 I以大写或小写字e d入b些值。
+引号	+引号m >在窗Z中4 = D信息。如在仿f器会话Y作员信息x r （OIA）中v现D信息。
例子类型	例子类型m >要s您在｜n a >或窗Z中d入D信息。

数V约定

二x制}	m >为 BX'xxxx xxxx' 或 BX'x', }非在某些i v下C文>m >（『二x制xxxx xxxx D值G...』）。

位位置	在最R位置（最无效位）I 0 * <。
. x 制 }	4 位 } 字以O D . x 制 } 以公制 = m >。 9 C U q 分 t 3 位一组D } 字，而；G C逗号。例如， } 16147 写I 16 147。
. y x 制 }	在文 > 中m > 为 . y x 制 xxxx 或 X'xxxx' （『相Z Z c D X址为 . y x 制 5D，指定为 X'5d'。』）

到何处 i 找更多E息

如需 | 多信息， k N阅 *l Y*入 *E*，此 i | 含P关通信服务器图 i b及其相关v f物D完{ h v。

在通信服务器2装后要 i 4某一 i 籍， k从您D桌 f 执行下P = 骤:

1. L序
2. IBM 通信服务器
3. 文5
4. 从 i 籍P m中选择

通信服务器 i 籍以 Portable Document Format（PDF）q = 存在， I C Adobe Acrobat Reader i 4。如果您D机器中；P此L序D = 4，则I从文5 P m中x行2装。

Internet O通信服务器D主页中P一c D z品信息，以及P关 APAR 和修 } D服务信息。要C = C主页， k 9 C某一 Internet / 览器，例如 IBM Web Explorer，然后转至下P URL:

http://www.software.ibm.com/enetwork/commserver/about/csnt.html

如需 | 多信息， k N阅 *l Y*入 *E*，此 i | 含P关v人通信b及其相关v f物D完{ h v。

在v人通信2装后要 i 4某一 i 籍， k从您D桌 f 执行下P = 骤:

1. L序
2. IBM 通信服务器
3. 文5
4. 从 i 籍P m中选择

v人通信 i 籍以 BookManager（BOO）q = 存在， I C IBM Library Reader i 4。如果您D机器中；P此L序D = 4，则I从 eNetwork v人通信 CD-ROM 中x行2装。

Internet O v人通信D主页中P一c D z品信息，以及P关 APAR 和修 } D服务信息。要C = C主页， k 9 C某一 Internet / 览器，例如 IBM Web Explorer，然后转至下P URL:

http://www.software.ibm.com/enetwork/pcomm/

第1？分 个人通E和通E服务器Z点Y w员h施

© Copyright IBM Corp. 1989, 1997, 1998

第1章 i \

> ? 分 h v v 人通信和通信服务器 y a 供 D Z c Y 作员 h)（NOF）API。

文档的 C 途

文 5 D 目 j G:

- a 供 Z c Y 作员 h) API a 构 D E v
- 定义在 S Z 中 w 动 D 信号 D o 法。

个人通 E 和通 E 服务器 Z 点 Y w 员 h 施

v 人通信和通信服务器 Z c Y 作员 h) 支 V Z c Y 作员、X 制 c（CP）和 _ 辑 % 元（LU）之间 D 通信。Z c Y 作员 h) 从 Y 作员处 S U Z c 配置信息，Y 作员在 Z c 启动 1 C 此信息来 u < 化 X 制 c。Z c Y 作员 h) 还 S U k s，以 i 询和显 > Z c 配置信息。Z c Y 作员能:

- 定义及 > } LU、DLC、端 Z 和 4 7
- 激活及 M 放 4 7 和会话
- i 询 X 制 c 和 LU D }] b 和状，信息

Z c Y 作员 I 为一 v 9 C；互 = 显 > D Y 作 _，一 v；文件 S Z 访问 D | n 文件，或一 v B 务 L 序。Z c Y 作员 h) 通过 9 C 动词 S Z k Z c Y 作员 x 行通信。

入口点

v 人通信和通信服务器 a 供 K 处理 Z c Y 作员 h) 动词 D b 文件。

Z c Y 作员 h) 动词 P 直 S D o 言 g f。您 D L 序 <] F 为*动词 X 制 i* D 内存 i 中 D 字段。然后，L 序 w C 入 Z c " 传] 指 k 至动词 X 制 i。1 Y 作完 I 1，Z c Y 作员 h) 返回已 9 C 和修 D 动词 X 制 i 中 D 字段。L 序 M I 以读取从动词 X 制 i 返回 D N }。

下 P G Z c Y 作员 h) 动词 D 入 Z c P m:

- WinAsyncNOF()
- WinAsyncNOFEx()
- WinNOFCancelAsyncRequest()
- WinNOFCleanup()
- WinNOFStartup()
- WinNOFRegisterIndicationSink()
- WinNOFUnregisterIndicationSink()
- WinNOFGetIndication()

k N 阅 Z 3 B Z c Y 作员 h) 入 Z c 以获取入 Z c D 详细 5 w。

© Copyright IBM Corp. 1989, 1997, 1998

动词控制块（VCB）

程 r h F " b: 基 > Y作系统通过在w C & C L 序D X址U间中执行一些子系统来E化性能。 b意味着& C L 序对 > X h v 符m（LDT）选择器D ； } 确 9 C + < 致； j 准D Y作，也I能引起系统 ' \ 。相& D，& C L 序；& C执行N k | D指k D LDT 选择器字段D指 k c u 运 c 。

动词X制 i (VCB) y 9 C D段X须G I 读I 写D }] 段。 L 序或 _ Q VCB y w为L 序中D一 v d ？来分配 |，或 _ G从一 v O大D段再次分配 |。 | X须大 = 足以 | 含L 序 } 在发 v D动词D y P 字段。

& C L 序；& C在动词X制 i 发 v = 动词完 I 之间， | D动词X制 i D任何？分。 1 Z c Y作员 h) a x 动词D执行后， | + 一 v 完 { D、修D过D VCB = 4 = 原来D i O。 y以，如果L 序Q动词X制 i 5 w为一 v d ？， k 在 2 ，存储器中 5 w，而；G 在内？过L D堆；中 5 w。

C 0(X'00') n d ？ v VCB 中D y P # t 和未 9 C D字段。 B 5 O，在L 序 3 值 x N } 之 0 h 置 { v 动词X制 i 为 0 也许在 1 间O | 为P 效。 h 置 # t D字段为 0 X p 重要。

"： 如果 VCB ； I 读写，或如果 | ；足 10 v 字Z（即，大 = 足够容纳Z c Y作员 h) D主返回 k 和次返回 k），则Z c Y作员 h) ；能访问 |，而R基 > Y作系统 + 异 # 终止 x L。 b 种终止 j 6 为 *# 规 # 护故 O*，处理器异 # i v 陷 e D。

1 VCB 过短或 9 C K ； } 确 D 段类型 1， Z c Y 作员 h) + 返回 INVALID_VERB_SEGMENT 主返回 k 。

编 4 Z 点Y w 员 h 施（NOF）程 r

v 人通信和通信服务器 a 供处理 NOF 动词D动，4 S b（DLL）文件。

DLL G I 重入D; 多 v & C L 序 x L 和线L I 以同 1 w C DLL。

NOF 动词P 直S D o 言 g f 。您D L 序 <] F 为*动词X制 i* (VCB)D内存 i 中D字段。然后， | w C NOF DLL " 传] 一 v 指 k = 动词X制 i 。 C Y作完 I 后， NOF 返回已 9 C和修D VCB 中D字段。 L 序M I 以读取从动词X制 i 返回D N } 。

Z 6页D m 1 显 > y a 供头文件D源L 序模 i C法，以及 ` 译和 4 S NOF L 序 y 需D b 。某些头文件 I 能 | 括其 | y 需D头文件。

m 1. NOF D头文件和 b

Y w 系统	头文 ~	库	DLL 名称
WINNT & WIN95	WINNOF.H	WINNOF32.LIB	WINNOF32.DLL
WIN3.1	WINNOF.H	WINNOF.LIB	WINNOF.DLL
OS/2	APPC_C.H	APPC.LIB	APPC.DLL

通E服务器 SNA API 客户机支持

此信息v J C Z通信服务器。

通信服务器|括一系P基Z Windows 95、Windows NT、Windows 3.1 和 OS/2 Y作系统DM户机。 b些M户机在> i中F为 SNA API M户机，" R v支V全？Z c Y作员h) D一v子集。 _ e X，**WINNOF** G唯一在 Windows M户机（95、NT、3.1）O支V D API。下P G y支V NOF D P m:

- QUERY_LOCAL_LU
- QUERY_LU_0_TO_3
- QUERY_LU_POOL
- QUERY_MODE
- QUERY_MODE_DEFINITION
- QUERY_PARTNER_LU
- QUERY_PARTNER_LU_DEFINITION
- QUERY_PU
- QUERY_SESSION
- QUERY_TP
- QUERY_TP_DEFINITION

通E服务器支持而个人通E ；支持的动词

此信息v J C Z通信服务器。

下P动词；通信服务器支V，而；；v人通信支V。

- DEFINE_DOWNSTREAM_LU
- DEFINE_DOWNSTREAM_LU_RANGE
- DEFINE_DSPU_TEMPLATE
- DELETE_DOWNSTREAM_LU
- DELETE_DOWNSTREAM_LU_RANGE
- DELETE_DSPU_TEMPLATE
- QUERY_ADJACENT_NN
- QUERY_DIRECTORY_STATS
- QUERY_DOWNSTREAM_LU
- QUERY_DOWNSTREAM_PU
- QUERY_DSPU_TEMPLATE
- QUERY_HPR_STATS
- QUERY_ISR_SESSION
- QUERY_NN_TOPOLOGY_NODE
- QUERY_NN_TOPOLOGY_STATS

- QUERY_NN_TOPOLOGY_TG
- DOWNSTREAM_LU_INDICATION
- DOWNSTREAM_PU_INDICATION
- ISR_INDICATION
- NN_TOPOLOGY_NODE_INDICATION
- NN_TOPOLOGY_TG_INDICATION

第2章 本书中动词的概述

> i 中 h v D动词 S Z 9 您D L 序能执行 k v 人通信或通信服务器网 g 环 3 关 * D大多 } 配置、系统管理和 Z c 定义功能。 > B a 供 ? 一功能和关 * 动词D 一 v E v。

如何阅读动词 5 明

Z 4 = 12 B h v 配置、系统管理和 , S 管理器动词。

提供N数

? v 动词P 一 v ? 分, a 供 P 关 I L 序 a 供N } 和任何关 * N } 值D详细 5 w。

在某些 i v 下, 您 I 以为一 v N } a 供一 v d ? 值。

返回N数

? v 动词P 一 v ? 分, a 供 P 关返回 x L 序D N } 和任何关 * N } 值D详细 5 w。

返回 k

> i 中 h v D配置、系统管理和 , S 管理器动词 _ P k 其关 * D返回 k, a 供 P 关动词 I 功执行D信息, 或 _ a 供错误信息。 b 些代 k 在 ? v 动词D 『返回N } 』 ? 分 P v。

附S E息

许多动词 5 w还 | 含一 v j b 为 『 = 加信息 』 D ? 分。 C ? 分 a 供 k 动词 P 关D其 { P C信息。

公共 VCB V段

> B h v 在Z c Y作员 h) API 中传] D ? v 动词D o 法。 | 还 h v 在 ? v 动词中传] 及返回 x 动词D N }。

```
typedef struct nof_hdr
{
        unsigned short  opcode;
        unsigned char   reserv2;            /* reserved */
        unsigned char   format;
        unsigned short  primary_rc;
        unsigned long   secondary_rc;
} NOF_HDR;
```

? v VCB P 多 v 公共字段。 P v 及 h v 如下。

opcode

动词Y作 k。此字段 j 6 动词。

format

j 6 VCB D q =。此字段X须 h 置为 (以指定 VCB D 1 0 f >) D值在 ? v 动词下%独 h v。

© Copyright IBM Corp. 1989, 1997, 1998

primary_rc

主返回 k 。 ？ v 动词D I 能值在？ v 动词？分中P v 。

secondary_rc

次返回 k 。 9 d I 主返回 k a 供D信息。

动词摘 *

Z c Y作员 h ） API I I C Z 执行下P 任务D动词组 I :

- 配置Z c 资源
- 激活及M放 4 7 和会话
- i 询 I Z c # V D信息
- | D会话 }
- 处理主动 a 供D指 >
- a 供Z n 支V
- 『ping』 一 v 远L LU
- 定义、 i 询和 > } CPI-C 方信息

Z 点配置

下P 动词 I C Z 定义资源:

- DEFINE_ADJACENT_NODE
- DEFINE_CN
- DEFINE_COS
- DEFINE_DEFAULT_PU
- DEFINE_DLC
- DEFINE_DLUR_DEFAULTS
- DEFINE_DOWNSTREAM_LU

DEFINE_DOWNSTREAM_LU v J C Z 通信服务器。

- DEFINE_DOWNSTREAM_LU_RANGE

DEFINE_DOWNSTREAM_LU_RANGE v J C Z 通信服务器。

- DEFINE_DSPU_TEMPLATE
- DEFINE_FOCAL_POINT
- DEFINE_INTERNAL_PU
- DEFINE_LOCAL_LU
- DEFINE_LS
- DEFINE_LU62_TIMEOUT
- DEFINE_LU_0_TO_3
- DEFINE_LU_0_TO_3_RANGE

- DEFINE_LU_POOL
- DEFINE_MODE
- DEFINE_PARTNER_LU
- DEFINE_PORT
- DEFINE_TP

下 P 动词 I C Z > } 资源:

- DELETE_ADJACENT_NODE
- DELETE_CN
- DELETE_COS
- DELETE_DLC
- DELETE_DOWNSTREAM_LU

DELETE_DOWNSTREAM_LU v J C Z通信服务器。

- DELETE_DOWNSTREAM_LU_RANGE

DELETE_DOWNSTREAM_LU_RANGE v J C Z通信服务器。

- DELETE_DSPU_TEMPLATE
- DELETE_FOCAL_POINT
- DELETE_INTERNAL_PU
- DELETE_LOCAL_LU
- DELETE_LS
- DELETE_LU62_TIMEOUT
- DELETE_LU_0_TO_3
- DELETE_LU_0_TO_3_RANGE
- DELETE_LU_POOL
- DELETE_MODE
- DELETE_PARTNER_LU
- DELETE_PORT
- DELETE_TP

$ 活和释放

下 P 动词在 4 7 c 9 C:

- START_DLC
- START_LS
- START_PORT
- STOP_DLC
- STOP_LS
- STOP_PORT

下 P 动词 C Z 从 t LU k s 器功能:

- START_INTERNAL_PU
- STOP_INTERNAL_PU

下 P 动词在会话 c 9 C:

- ACTIVATE_SESSION
- DEACTIVATE_CONV_GROUP
- DEACTIVATE_SESSION

下 P 动词 C Z ? 制一 v 至；换 7 6 D _ 性能 7 I 选择（HPR）RTP ，S:

PATH_SWITCH

i 询 Z 点

b 些动词在 _ { 字段中返回 Z c 信息:

- QUERY_DEFAULT_PU
- QUERY_DLUR_DEFAULTS
- QUERY_MDS_STATISTICS
- QUERY_NN_TOPOLOGY_STATS

QUERY_NN_TOPOLOGY_STATS v J C Z 通信服务器。

- QUERY_NODE
- QUERY_STATISTICS

下 P 动词 I 返回一 v 或多 v 信息%元:

- QUERY_ADJACENT_NN
- QUERY_ADJACENT_NODE
- QUERY_CN
- QUERY_CN_PORT
- QUERY_COS
- QUERY_DEFAULTS
- QUERY_DLUS
- QUERY_DOWNSTREAM_PU

QUERY_DOWNSTREAM_PU v J C Z 通信服务器。

- QUERY_DSPU_TEMPLATE
- QUERY_FOCAL_POINT
- QUERY_LU_POOL
- QUERY_LU62_TIMEOUT
- QUERY_MDS_APPLICATION
- QUERY_MODE_TO_COS_MAPPING
- QUERY_NMVT_APPLICATION

- QUERY_PU
- QUERY_TP

I 认为此信息G以P m形=存储D。动词I在P m中指定一v _ {项，而b一项+；认为G一v位置j记（或w引值）。b些动词中D **list_options** 字段指定信息从P m中D哪一c返回。

- 如果 **list_options** h置为 AP_FIRST_IN_LIST，则指定w引值D字段+；忽T，"R返回P m+从P m D*头*<。
- 如果 **list_options** h置为 AP_LIST_INCLUSIVE，则返回P m+从指定w引值*<。
- 如果 **list_options** h置为 AP_LIST_FROM_NEXT，则返回P m+从指定w引值后D项*<。

w引值指定返回信息D起<c。w引值确定之后，一些i询动词还a供返回P m D其{过K选项。b些选项D指定独"Z w引值。注意，}非指定，否则返回P m+y] IBM D 6611 APPN MIB x行排序。（P关动词N}如何3 d= MIB m q项D信息，k N阅Z615页D『=<A. IBM APPN MIB m』。）

要返回D项D}目或要n d D缓e x大小已h置。（如果=_都已h置，则动词+返回信息D=v指定}？中O小D那v。）因为&C L序缓e x大小通#会限制I返回信息D}？，y以Z c Y作员h）+返回=加D信息，以指>返回k s信息需要D缓e x大小总}，以及|y m>D总项}。

}K返回一v或多v信息%元外，下P动词还能返回；同级p D信息。通过在 **list_options** 字段中|含 AP_DETAIL 或 AP_SUMMARY， **list_options** 字段I指定返回D G*要信息还G详细信息。b些选项通过 **OR**（**或**Y w）以0 **list_options** 中D一v来指定，例如: AP_DETAIL | AP_FIRST_IN_LIST。

- QUERY_DIRECTORY_LU
- QUERY_DLC
- QUERY_DLUR_LU
- QUERY_DLUR_PU
- QUERY_DOWNSTREAM_LU

QUERY_DOWNSTREAM_LU v J C Z通信服务器。

- QUERY_ISR_SESSION

QUERY_ISR_SESSION v J C Z通信服务器。

- QUERY_LOCAL_LU
- QUERY_LOCAL_TOPOLOGY
- QUERY_LS
- QUERY_LU_0_TO_3
- QUERY_MODE
- QUERY_MODE_DEFINITION
- QUERY_NN_TOPOLOGY_NODE

QUERY_NN_TOPOLOGY_NODE v J C Z通信服务器。

- QUERY_NN_TOPOLOGY_TG

QUERY_NN_TOPOLOGY_TG v J C Z通信服务器。

- QUERY_PARTNER_LU
- QUERY_PARTNER_LU_DEFINITION
- QUERY_PORT
- QUERY_RTP_CONNECTION
- QUERY_SESSION
- QUERY_TP_DEFINITION

会话限制动词

- CHANGE_SESSION_LIMIT
- INITIALIZE_SESSION_LIMIT
- RESET_SESSION_LIMIT

主动提供的指示

显 > Z c 信息D & C L 序 I 9 C b 些指 > （ v 1 发 z | D " 返回 * 要信息 1 发 v ）来触发 i 询动词（返回详细信息）。如 P 任何 & C L 序注 a S U 信息， Z c + v z z 在以下 P v D 信号，作为 _ { B 件D主动 a 供D指 > 。 因此，如果 & C L 序；再需要信息，& 取消注 a 。

- DLC_INDICATION
- DLUR_LU_INDICATION
- DLUS_INDICATION
- DOWNSTREAM_LU_INDICATION

DOWNSTREAM_LU_INDICATION v J C Z通信服务器。

- DOWNSTREAM_PU_INDICATION

DOWNSTREAM_PU_INDICATION v J C Z通信服务器。

- FOCAL_POINT_INDICATION
- ISR_INDICATION

ISR_INDICATION v J C Z通信服务器。

- LOCAL_LU_INDICATION
- LOCAL_TOPOLOGY_INDICATION
- LS_INDICATION

- LU_0_TO_3_INDICATION
- MODE_INDICATION
- NN_TOPOLOGY_NODE_INDICATION

NN_TOPOLOGY_NODE_INDICATION v J C Z通信服务器。

- NN_TOPOLOGY_TG_INDICATION

NN_TOPOLOGY_TG_INDICATION v J C Z通信服务器。

- PLU_INDICATION
- PORT_INDICATION
- PU_INDICATION
- REGISTRATION_FAILURE
- RTP_INDICATION
- SESSION_INDICATION
- SESSION_FAILURE_INDICATION

作为指 > 9 C D入Z c G:

WinNOFRegisterIndicationSink
注 a 以 S U指 >

WinNOFUnregisterIndicationSink
取消 S U指 > D注 a

WinNOFGetIndication
S U指 >

b 些指 > 传] = y P C Z c Y作员 h) 注 a D指 > S U器中。如果 z I 指 > D ? 件；能发M C指 >，则 | + 在其发M D下一 v 指 > 中 h 置 **data_lost** 指 > 符。 如果在某一指 > O D **data_lost** j 志位置为 AP_YES，则后继 }] 字段 I h 置为U。此 j 志C Z通知& C L 序发 z K 一 v | D，而其详细信息已丢 '， | 指 > & C L 序&通过发 v J 1 D i 询动词来作 v 响&。

注意，信号 LULU_EVENT 也；归类为一 v 指 >，因为 | G I Z c 主动发M x 一 v C动词 REGISTER_LULU_EVENT 和 UNREGISTER_LULU_EVENT 注 a D x L。 | 未在 O f P v，原因G其习性P显著 n 异: 注 a C Z 一 v LU-伙 i LU 对，R无 **data_lost** D H价物 -- b 些 LULU B件指 > I 功 z I。

2 全 T 动词

下 P 2 全性动词允许 LU-LU 验证或对话 2 全性D Z n 管理。

- DEFINE_LU_LU_PASSWORD
- DEFINE_USERID_PASSWORD
- DELETE_LU_LU_PASSWORD
- DELETE_USERID_PASSWORD

APING 动词

下 P 动词允许一 v 管理& C L 序 『ping』 一 v 网 g 中 D 远 L LU。

APING

CPI-C 动词

下 P 动词允许定义、 i 询和 > } CPI-C 方信息。

- DEFINE_CPIC_SIDE_INFO
- DELETE_CPIC_SIDE_INFO
- QUERY_CPIC_SIDE_INFO

如需 P 关 v 人通信和通信服务器 Windows 95 和 Windows NT f y a 供 CPI-C 支 V D | 多信息, k N 阅 *CPI-C Reference*。

，S 管 m 器动词

下 P 动词 I C Z X 制，S 管理器:

- DISABLE_ATTACH_MANAGER
- ENABLE_ATTACH_MANAGER
- QUERY_ATTACH_MANAGER

DLC 过程、端口和 4 7 站

DLC 过程

v 人通信或通信服务器 I 创 (多 v DLC 过 L。？v DLC 过 L I v 人通信或通信服务器创 (, 以响&在 Z c Y 作员 h) API 处发 v D START_DLC 动词。？v DLC ：责一 v 4 7 或一组 4 7 O D 通信, b 些 4 7 9 C 某一 X 定 }] 4 7 协议 (例如 SDLC 或 n 牌环)。

？v DLC 过 L I 管理一 v 或多 v 端 Z。端 Z 如下 y v。

端口

端 Z m > 一 v 在 > X 机器中唯一 D 存取 c （例如一 v MAC/SAP X 址对）, " k 一 v DLC 过 L 关 *。？v DLC I P 一 v 或多 v 端 Z。一 v 端 Z I 为下 P 类型之一:

；换式端口

I P 一 v 或多 v 在任何 1 L 都活动 D 相 Z 4 7 >。(注意, b k *SNA APPN Architecture Reference* 中 D 定义；同。)

非；换式端口

I P c = c 4 S 和多 c 4 S。非；换 = 4 7 O D 相 Z 4 7 > X 须 C Z c Y 作员 h) ? 件 x 行定义。多 c 非；换 = 4 7 要 s 在 y P Z c O } 确定义主/从关系, 以 \ b ； I 预 b D a 果。

SATF 端口

9 C 一 v 共享访问 D 传 M h), 例如 n 牌环。 | 允许 k h) , S D 任何 4 7 >

对之间D，通性。 y P在n牌环O激活D 4 7 > D u < G + X须<终定义为I协L， b样4 7激活M I通过任何4 7 >启动。

": SATF 端Z还I k ，S网g关*。在b种i v下， C拓扑 | 新来广%唯一存取c D X址。

4 7站

一v 4 7 > k某一端Z关*， " m >至某一相Z Z c D ，S。一v端Z I P多v 4 7 >。 4 7 > I 4下P方=分类:

Q定e 4 7站

已显=定义D 4 7 >（ 9 C DEFINE_LS 动词)。

动态4 7站

因激活，S网g中D某一动，，S而创(D 4 7 >（也F为虚拟7 I选择Z c（VRN））。

~式4 7站

因从某一以0未知伙i Z c（在；换=端Z或 SATF 端Z O ）S U =一v呼P而创(D 4 7 >。(此类型端Z未在 *SNA APPN Architecture Reference* 中定义。)

Y时4 7站

1在 DLC S Z（在；换=端Z或 SATF 端Z O ）O S U =一v CONNECT_IN 1创(D 4 7 >。 1远L Z c j 6确定1， |要4 ；> }，要4 d为动，或隐=4 7 >。

第3章 Z点Y w员h施入口点

>B h v Z c Y作员h)动词D人Z c。

© Copyright IBM Corp. 1989, 1997, 1998

WinNOF()

此函 } 为 y P Z c Y作员 h) 动词 a 供一 v 同 = 入 Z c 。

o 法

```
void WINAPI WinNOF(long vcb,
                   unsigned short vcb_size)
```

N数　5明

vcb　指向动词X制 i D指 k 。

vcb_size
动词X制 i 中D字 Z } 。

返回

无返回值。动词X制 i 中D **primary_rc** 和 **secondary_rc** 字段指 v 某些错误。

备 "

b G Z c Y作员 h) API D主同 = 入 Z c 。直 = 动词完 I E w C i 。

WinAsyncNOF()

此函 } 为 y P Z c Y作员 h) 动词 a 供一 v 异 = 入 Z c。

o 法

```
HANDLE WINAPI WinAsyncNOF(HWND hwnd,
                          long vcb,
                          unsigned short vcb_size)
```

N数　5 明

hwnd　S U完 I 消息D窗 Z d z。

vcb　指向动词X制 i D指 k。

vcb_size
　　动词X制 i 中D字 Z }。

返回

返回值指定异 = k s G否 I 功。如果函 } I 功，5 际D返回值 + G一 v d z。如果函 } ; I 功，则返回 c。

备 "

在 9 C此入 Z c 1，？v & C L 序线 L 在同一 1 L 只能 P 一 v 未完 I k s。

1 异 = Y作完 I 1，& C L 序D窗 Z *hWnd* S U带 **"WinAsyncNOF"** 作为 d 入字符串返回D **RegisterWindowMessage** 消息。*wParam* d ? | 含 I 原 < 函 } w C返回D异 = 任务 d z。

如果函 } I 功返回，则 1 Y作完 I 或对话取消 1 + 发M一 v **WinAsyncNOF()** 消息 = & C L 序。

"**:** m见 **WinNOFCancelAsyncRequest()**。

WinAsyncNOFEx()

此函 } 为 y P Z c Y 作员 h) 动词 a 供一 v 异 = 入 Z c 。 9 C 此入 Z c 代 f i w C，以允许在同一线 L O 处理多 v 动词。

o 法

```
HANDLE WINAPI WinAsyncNOFEx(HANDLE handle,
                            long vcb,
                            unsigned short vcb_size);
```

N数　5 明

handle

& C L 序 + H 待 D B 件 D d z 。

vcb　指向动词 X 制 i D 指 k 。

vcb_size

动词 X 制 i 中 D 字 Z } 。

返回

返回值指定异 = k s G 否 I 功。如果函 } I 功， 5 际 D 返回值 + G 一 v d z 。

备 "

此入 Z c 意在 k Win32** API 中 D WaitForMultipleObjects 一起 9 C。如需 P 关此函 } D | 多信息， k N 阅 Win32 API D L 序 h 计文 5 。

1 异 = Y 作完 I 1 ，通过发 B 件 D 信号通知 & C L 序。在发 B 件 D 信号 1 ， + 检 i 主返回 k 和次返回 k D 任何错误 u 件。

"： m 见 **WinNOFCancelAsyncRequest()**。

WinNOFCancelAsyncRequest()

此函 } 取消一 v 基 Z k s D未完 I **WinAsyncNOF**。

o 法

```
int WINAPI WinNOFCancelAsyncRequest(HANDLE handle);
```

N数 5明

handle
a 供N }； 指定要取消 k s D d z。

返回

返回值指定异 = k s G否取消。 如果值为 c，则 k s ；取消。否则，值为:

WNOFALREADY
一 v 错误代 k，指 > ；取消 D 异 = k s 已完 I，或 d z 无效。

备 "

I 某一 **WinAsyncNOF** 函 } 在以 0 发 v D 异 = k s I 在完 I 之 0 ；取消，方法 G 发 v **WinNOFCancelAsyncRequest()** w C，指定 I u < 函 } 在 *handle* 中返回 D d z。

取消一 v 异 = k s + 停止任何对 & C L 序动词 X 制 i D | 新，" 停止； f 知动词已完 I D & C L 序（通过窗 Z 消息或 B 件）。；取消基 > k s。要 5 际 X 取消基 > k s，& C L 序 X 须发 v J 1 D NOF 动词（即，发 v STOP_LS 以取消 START_LS）。

如果一 v 取消某一现 P 异 = **WinAsyncNOF** 例行 L 序 D " T ' \，错误 k 为 WNOFALREADY，则 & 发 z K 下 P = _ 之一。或 _ 原 < 例行 L 序已完 I， R & C L 序处理 a 果通知；或 _ 原 < 例行 L 序已完 I， + & C L 序；处理完 I 通知。

"： m 见 **WinAsyncNOF()**。

WinNOFCleanup()

此函 } 从 Z c Y 作员 h ） API 终止 " 注销 & C L 序。

o 法

```
BOOL WINAPI WinNOFCleanup(void);
```

返回

返回值指定注销 G 否 I 功。如果值；为 0，则 & C L 序 I 功注销。如果返回值为 0，则 & C L 序未注销。

备 "

9 C **WinNOFCleanup()** 来指 > 从 Z c Y 作员 h ） API 注销一 v Z c Y 作员 h ）& C L 序。

WinNOFCleanup 对任何在 **WinNOFGetIndication** 中 H 待 D 线 L b } 阻塞。带 P WNOFNOTREG 返回（& C L 序；注 a 来 S U 指 > ）。**WinNOFCleanup** 为 y P 指 > 注销 & C L 序。**WinNOFCleanup** 返回某些带 P 错误 AP_CANCELLED D 未完 I 动词（同 = 或异 = ）。然而，动词 G 在 Z c 内完 I D。

9 C **WinNOFStartup** 和 **WinNOFCleanup** " 非 X 需。然而，& C L 序 X 须 k 其 9 C b 些 w C 相一致。您 & C = v 都 9 C 或 = v 都；9 C。

"： m 见 **WinNOFStartup()**。

WinNOFStartup()

此函 } 允许 & C L 序指定 y 需 Z c Y 作员 h ） API D f >, " 检 w z 品 y 支 V API D f >。此函 } I 在发 v 任何 x 一 = D Z c Y 作员 h ） API w C 以注 a 之 0 ；& C L 序来 w C。

o 法

```
int WINAPI WinNOFStartup(WORD wVersionRequired,
                         LPWNOFDATA nofdata);
```

N 数　5 明

wVersionRequired

指定要 s D Z c Y 作员 h ） API 支 V D f >。 _ 位字 Z 指定次 f >(revision)号; M 位字 Z 指定主 f > 号。

nofdata

返回 Z c Y 作员 h ） API D f > 以及 API 5 现 D 一 v 5 w。

返回

返回值指定 & C L 序 G 否已 I 功注 a，以及 Z c Y 作员 h ） API 5 现 G 否 I 支 V 指定 D f > 号。如果值为 0，则为注 a I 功 R I 以支 V 指定 D f >。否则，返回值 G 下 P 值之一:

WNOFSYSERROR

基 > 网 g 子系统未准 8 好 x 行网 g 通信。

WNOFVERNOTSUPPORTED

此 X 定 5 现未 a 供 y k s Z c Y 作员 h ） API 支 V D f >。

WNOFBADPOINTER

；} 确 D nofdata N } 。

备 "

此 w C 意在对 API + 来发行 > D 兼容性 a 供 o 助。 1 0 f > 为 1.0。

9 C **WinNOFStartup** 和 **WinNOFCleanup** " 非 X 需。然而，& C L 序 X 须 k 其 9 C b 些 w C 相一致。您 & C = v 都 9 C 或 = v 都；9 C。

"： m 见 **WinNOFCleanup()**。

WinNOFRegisterIndicationSink()

| 允许& C L 序注 a 以 S U 非 k s D 指 > 。

o 法

```
BOOL WINAPI WinNOFRegisterIndicationSink(unsigned short indication_opcode,
                                         unsigned short queue_size,
                                         unsigned short *primary_rc,
                                         unsigned long *secondary_rc);
```

N数　5 明

indication_opcode
要注 a D 指 > 。

queue_size
未U = 排队指 > D次 } 。 c m > 9 C 1 0 值（ u < 缺！值 h 置为 10）。 y P I & C L 序注 a D 指 > 只 P 一 v 队 P 。

primary_rc
返回: 主要返回 k

secondary_rc
返回: 次级返回 k

返回

函 } 返回一 v 指 > 注 a G 否 I 功 D 值。如果值；为 0, 则注 a I 功。如果值为 0, 则注 a ； I 功。

备 "

9 C **WinNOFRegisterIndicationSink** 来注 a S U **indication_opcode** 类型 D 非 k s 指 > 。

& C L 序对 Z ？种要 S U D 指 > 类型 X 须发 v 一 v **WinNOFRegisterIndicationSink**。

"： m 见 **WinNOFUnregisterIndicationSink** 和 **WinNOFGetIndication**。

WinNOFUnregisterIndicationSink()

| 允许& C L 序停止 S U非 k s D指 > 。

o 法

```
BOOL WINAPI WinNOFUnregisterIndicationSink(unsigned short indication_opcode,
                                           unsigned short *primary_rc,
                                           unsigned long *secondary_rc);
```

N数　5 明

indication_opcode
要注销D指 > 。

primary_rc
返回: 主返回 k 。

secondary_rc
返回: 次返回 k 。

返回

函 } 返回一 v 指 > 注销G否 I 功D值。如果值；为 0, 则注销 I 功。 如果值为 0, 则注销； I 功。

备 "

9 C **WinNOFUnregisterIndicationSink** 来停止S U **indication_opcode** 类型D非 k s 指 > 。

& C L 序对 Z ? 种要停止 S U D 指 > 类型 X 须发 v 一 v **WinNOFUnregisterIndicationSink**。

"**:** m见 **WinNOFRegisterIndicationSink** 和 **WinNOFGetIndication**。

WinNOFGetIndication()

b 允许& C L 序 S U非 k s D指 > 。

o 法

```
int WINAPI WinNOFGetIndication(long buffer,
                               unsigned short *buffer_size,
                               unsigned long timeout);
```

N数　5明

buffer 指向 S U指 > 缓 e x D指 k 。

buffer_size
缓 e x 大小。返回: 指 > D大小。

timeout
以毫 k 为%位D H待指 > 1 间。

返回

函 } 返回一 v 值指 > G否 S U = 指 > 。

0 返回D指 > 。

WNOFTIMEOUT
H待指 > D ， 1 。

WNOFSYSNOTREADY
基 > 网 g 子系统未准 8 好 x 行网 g 通信。

WNOFNOTREG
& C L 序 ; 注 a 来 S U指 > 。

WNOFBADSIZE
缓 e x + 小， ; 能 S U 指 > 。 在 P 足够大 D 缓 e x 1 ， 再次发 v **WinNOFGetIndication** w C。指 > D大小返回 = **buffer_size** N } 中。

WNOFBADPOINTER
缓 e x 或 **buffer_size** N } 无效。

WNOFSYSERROR
发 z 未预O = D系统错误。

备 "

b G一 v i w C， | 在下 P i v 之一返回:

- 返回一 v 指 >
- ， 1 = 期
- & C L 序发 v 一 v WinNOFCleanup w C
- z 品停止K
- 发 z 系统错误

"： m见 **WinNOFRegisterIndicationSink** 和 **WinNOFUnregisterIndicationSink**。

WinNOFGetIndication()

第4章 Z点配置动词

下P动词C Z定义和> } Z c配置信息。

© Copyright IBM Corp. 1989, 1997, 1998

DEFINE_ADJACENT_NODE

DEFINE_ADJACENT_NODE + d 入项m加= Z c 目< }] b中， b些d入项存放相Z Z c O D资源。

": 如果P一u至相Z Z c 9 C CP-CP 会话D活动7 6，则；需要此动词， " R ；&发v。

1 Z c D X制c ；m加= y目<中1，此动词I在端Z c O发v。

要定义Z c D X制c LU， h置下P字段:

- 在 **cp_name** 字段中指定Z c D X制c { F。
- m加一v ADJACENT_NODE_LU a构，在 **fqlu_name** 字段中指定X制c { F。

Z c O D y P其| LU 都；作为C Z c D X制c D子女m加= 目<中。DEFINE_ADJACENT_NODE 也I C Z + LU 定义m加=现P Z c定义中。LU I通过发v DELETE_ADJACENT_NODE 动词，以相同D方= > }。如果动词在处理过L中 ' \，则y P新(D目<项+ ；> }，而目< # V动词发v 0 D状，。

VCB a构

DEFINE_ADJACENT_NODE 动词|含多v ADJACENT_NODE_LU 2 G。ADJACENT_NODE_LU a构，S在 DEFINE_ADJACENT_NODE a构D最后。

```
typedef struct define_adjacent_node
{
        unsigned short  opcode;              /* verb operation code     */
        unsigned char   reserv2;             /* reserved                */
        unsigned char   format;              /* format                  */
        unsigned short  primary_rc;          /* primary return code     */
        unsigned long   secondary_rc;        /* secondary return code   */
        unsigned char   cp_name[17];         /* CP name                 */
        unsigned char   description[RD_LEN]; /* resource description    */
        unsigned char   reserv3[19];         /* reserved                */
        unsigned short  num_of_lus;          /* number of LUs           */
} DEFINE_ADJACENT_NODE;

typedef struct adjacent_node_lu
{
        unsigned char   wildcard_lu;         /* wildcard LU name        */
                                             /* indicator               */
        unsigned char   fqlu_name[17];       /* fully qualified LU name */
        unsigned char   reserv1[6];          /* reserved                */
} ADJACENT_NODE_LU;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_ADJACENT_NODE

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

cp_name

相Z端Z c中X制c D全限定{。|& k在Z c D XID O发M（如果Z c支V

XID）D Z c { F匹配， " k在 DEFINE_LS O为至Z c D 4 7 D相Z X制c { F匹配。{ F $度为 17 v字Z， "以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）

description

资源5 w（返回= QUERY_DIRECTORY_LU O）。 | G一v在> X I 显> 字符集中D一v 16 字Z（非c）字符串。 y P 16 v字Z都P效。

num_of_lus

z在 DEFINE_ADJACENT_NODE VCB 之后D相Z LU 2 G }目。

adjacent_node_lu.wildcard_lu

m w指定D LU { F G否G一v通配符{ F（AP_YES 或 AP_NO）。

adjacent_node_lu.fqlu_name

要定义D LU { F。如果此{ F ；G全限定D，则假h为 CP { F D网g ID。{ F $度为 17 v字Z， "以 EBCDIC U q R n d。 | I一v或= v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）

1 **wildcard_lu** 为 TRUE 1，一v后z EBCDIC U q D ².² m > 一v全通配符（k任何对象匹配）。 y P D EBCDIC U q + k任何以 CP { F D网g ID * 头D对象匹配。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_CP_NAME

AP_INVALID_LU_NAME
AP_INVALID_WILDCARD_NAME

如果因状，错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_CP_NAME

AP_INVALID_LU_NAME

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

secondary_rc

AP_MEMORY_SHORTAGE

AP_DIRECTORY_FULL

DEFINE_CN

DEFINE_CN 定义，S网g（也F为虚拟7 I选择Z c或 VRN）。动词a供，S网g D网g全限定{ F，以及，S网g D传d组（TG）X性。 |还a供一E > X端Z { F D P m， b些端Z I访问此，S网g。

DEFINE_CN I C Z重定义现P D，S网g。X p X，通过发v m一v DEFINE_CN, I +新D端Z m加=访问，S网g D端Z P m中。（I通过发v DELETE_CN 动词，以相同D方= > }端Z。）

VCB a构

```
typedef struct define_cn
{
        unsigned short  opcode;          /* verb operation code        */
        unsigned char   attributes;      /* verb attributes            */
        unsigned char   reserv2;         /* reserved                   */
        unsigned char   format;          /* format                     */
        unsigned short  primary_rc;      /* primary return code        */
        unsigned long   secondary_rc;    /* secondary return code      */
        unsigned char   fqcn_name[17];   /* name of connection network */
        CN_DEF_DATA     def_data;        /* CN defined data            */
        unsigned char   port_name[8][8];
                                         /* port names                 */
} DEFINE_CN;

typedef struct cn_def_data
{
        unsigned char   description[RD_LEN];
                                         /* resource description       */
        unsigned char   num_ports;       /* number of ports on CN      */
        unsigned char   reserv1[16];     /* reserved                   */
        TG_DEFINED_CHARS tg_chars;       /* TG characteristics         */
} CN_DEF_DATA;

typedef struct tg_defined_chars
{
        unsigned char   effect_cap;      /* effective capacity         */
        unsigned char   reserve1[5];     /* reserved                   */
        unsigned char   connect_cost;    /* connection cost            */
        unsigned char   byte_cost;       /* byte cost                  */
        unsigned char   reserve2;        /* reserved                   */
        unsigned char   security;        /* security                   */
        unsigned char   prop_delay;      /* propagation delay          */
        unsigned char   modem_class;     /* modem class                */
        unsigned char   user_def_parm_1; /* user-defined parameter 1   */
        unsigned char   user_def_parm_2; /* user-defined parameter 2   */
        unsigned char   user_def_parm_3; /* user-defined parameter 3   */
} TG_DEFINED_CHARS;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_CN

attributes

动词D t性。此字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

ｊ 6 VCB D q = 。 + 此

def_data.tg_chars.prop_delay

传%延Y，m>一v信号在4 7中传MD 1间，%位为微k。此值`k为一v 1字ZD！c}，I公= 0.1mmm * 2 eeeee m>，其中字ZD位m>为eeeeemmm。缺！值如下y>:

AP_PROP_DELAY_MINIMUM

无传%延Y。

AP_PROP_DELAY_LAN

小Z 480 微k延Y。

AP_PROP_DELAY_TELEPHONE

在 480 和 49 512 微k间延Y。

AP_PROP_DELAY_PKT_SWITCHED_NET

在 49 512 和 245 760 微k间延Y。

AP_PROP_DELAY_SATELLITE

大Z 245 760 微k延Y。

AP_PROP_DELAY_MAXIMUM

最大传%延Y。

def_data.tg_chars.modem_class

t。此字段&<终h置为c。

def_data.tg_chars.user_def_parm_1

在 0--255 间DC户定义N}。

def_data.tg_chars.user_def_parm_2

在 0--255 间DC户定义N}。

def_data.tg_chars.user_def_parm_3

在 0--255 间DC户定义N}。

port_name

至多 8 v端Z{FD}组，b些端Z在，S网gO定义。?v_{端ZX须已I一v DEFINE_PORT 动词定义。?v端Z{FG一v在>XI显>字符集中D一v 8 字Z字符串，"RX须k在关* DEFINE_PORT 动词中D端Z{F匹配。通过发vm一v DEFINE_CN 指定新端Z{F，I在，S网gO定义其|端Z。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc

AP_OK

如果因N}错误而9动词；能执行，则L序返回下PN}:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_CN_NAME

AP_INVALID_NUM_PORTS_SPECIFIED
AP_INVALID_PORT_NAME
AP_INVALID_PORT_TYPE
AP_DEF_LINK_INVALID_SECURITY
AP_EXCEEDS_MAX_ALLOWED

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PORT_ACTIVE

AP_CANT_MODIFY_VISIBILITY

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_COS

DEFINE_COS m加一v服务级定义。DEFINE_COS 动词也ICZ修D一v以0定义COS 中D任何字段。

定义a供Zc和TG 『行』。 b些行+某一范围DZc和TG X性kCZ7I计cD权值相关*。权值越M, 7I越好。

VCB a构

DEFINE_COS 动词|含多v **cos_tg_row** 和 **cos_node_row** 2G。**cos_tg_row** a构，S在 DEFINE_COS D最后（"4}序权值排序），"R后z **cos_node_row** a构（也4}序权值排序)。

```
typedef struct define_cos
{
        unsigned short  opcode;           /* verb operation code         */
        unsigned char   reserv2;          /* reserved                    */
        unsigned char   format;           /* format                      */
        unsigned short  primary_rc;       /* primary return code         */
        unsigned long   secondary_rc;     /* secondary return code       */
        unsigned char   cos_name[8];      /* class-of-service name       */
        unsigned char   description[RD_LEN];
                                          /* resource description        */
        unsigned char   transmission_priority;
                                          /* transmission priority       */
        unsigned char   reserv3[9];       /* reserved                    */
        unsigned char   num_of_node_rows; /* number of node rows         */
        unsigned char   num_of_tg_rows;   /* number of TG rows           */
} DEFINE_COS;

typedef struct cos_node_row
{
        COS_NODE_STATUS minimum;          /* minimum                     */
        COS_NODE_STATUS maximum;          /* max                         */
        unsigned char   weight;           /* weight                      */
        unsigned char   reserv1;          /* reserved                    */
} COS_NODE_ROW;

typedef struct cos_node_status
{
        unsigned char   rar;              /* route additional resistance */
        unsigned char   status;           /* node status.                */
        unsigned char   reserv1[2];       /* reserved                    */
} COS_NODE_STATUS;

typedef struct cos_tg_row
{
        TG_DEFINED_CHARS minimum;         /* minimum                     */
        TG_DEFINED_CHARS maximum;         /* maximum                     */
        unsigned char    weight;          /* weight                      */
        unsigned char    reserv1;         /* reserved                    */
} COS_TG_ROW;

typedef struct tg_defined_chars
{
        unsigned char   effect_cap;       /* effective capacity          */
        unsigned char   reserve1[5];      /* reserved                    */
        unsigned char   connect_cost;     /* cost per connect time       */
        unsigned char   byte_cost;        /* cost per byte               */
        unsigned char   reserve2;         /* reserved                    */
        unsigned char   security;         /* security                    */
        unsigned char   prop_delay;       /* propagation delay           */
        unsigned char   modem_class;      /* modem class                 */
```

```
        unsigned char   user_def_parm_1;  /* user-defined parameter 1    */
        unsigned char   user_def_parm_2;  /* user-defined parameter 2    */
        unsigned char   user_def_parm_3;  /* user-defined parameter 3    */
} TG_DEFINED_CHARS;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_COS

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

cos_name

服务级{ F。b G一v 8字Z字母}字D A型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。

description

资源5 w（返回= QUERY_COS O）。| G一v在>X I显>字符集中D一v 16 字Z字符串。y P 16 v字Z都P效。

transmission_priority

传d E先级。| h置为下P值之一:

AP_LOW
AP_MEDIUM
AP_HIGH
AP_NETWORK

num_of_node_rows

z在 DEFINE_COS VCB 之后D Z c行2 G D }目。最大值G 8。？v Z c行|含一v最小Z c X性h置、一v最大Z c X性h置和一v权值。1计c一v Z c D权值1，对其X性k为？v Z c行定义D最小和最大X性x行对U检i。然后Z c；指定为Z一v Z c行D权值，C Z c行+y P D Z c X性限制在指定D限制内。如果Z c X性；z足任何P v D Z c行，则Z c；认为；J合C COS，"指定为一v无n大权值。注意，Z c行X须以权值D] } 3序x行"置。

num_of_tg_rows

z在Z c行2 G之后 TG 行2 G D }目。最大值G 8。？v TG 行|含一v最小 TG X性h置、一v最大 TG X性h置和一v权值。1计c一v TG D权值1，对其X性k为？v TG 行定义D最小和最大X性x行对U检i。然后 TG ；指定为Z一v TG 行D权值，C TG 行+y P D TG X性限制在指定D限制内。如果 TG X性；z足任何P v D TG 行，则 TG ；认为；J合C COS，"指定为一v无n大权值。注意，TG 行X须以权值D] } 3序x行"置。

cos_node_row.minimum.rar

7 I =加阻9最小值。值X须在 0--255 间。

cos_node_row.minimum.status

指定Z c D最小5塞状，。| I h置为下P值之一:

AP_UNCONGESTED

Z c无5塞。

AP_CONGESTED
ISR 会话D }目大Z **isr_sessions_upper_threshold**。

cos_node_row.maximum.rar
7 I = 加阻 9 最大值。值X须在 0--255 间。

cos_node_row.maximum.status
指定Z c D最大 5 塞状，。 | I h 置为下P值之一:

AP_UNCONGESTED
Z c 无 5 塞。

AP_CONGESTED
ISR 会话D }目大Z **isr_sessions_upper_threshold**。

cos_node_row.weight
k 此Z c 行关 * D权值。值X须在 0--255 间。

cos_tg_row.minimum.effect_cap
P 效能& 5 际%元D最小限制。此值 ` k 为一 v 1 字Z D ！ c }， I 公= 0.1mmm * 2 eeeee m >，其中字Z D位m >为 eeeeemmm。P效能&D ？ v %元H Z 300 位／ k 。

cos_tg_row.minimum.connect_cost
？，S 1 间D最小限制。P效值G 0--255 之间D { }值，其中 0 G最M？，S 1 间I >，而 255 G最 _ ？，S 1 间I >。

cos_tg_row.minimum.byte_cost
?字Z D最小限制。P效值G 0--255 之间D { }值，其中 0 G最M?字Z I >，而 255 G最 _ ?字Z I >。

cos_tg_row.minimum.security
2 全性值D最小极限如下 y >:

AP_SEC_NONSECURE
无 2 全性。

AP_SEC_PUBLIC_SWITCHED_NETWORK
在此，S网 g O发MD }] + 在公C ；换网 g O w动。

AP_SEC_UNDERGROUND_CABLE
}] 在 2 全S X g 缆中发M。

AP_SEC_SECURE_CONDUIT
线 7 G ；加 # 护D 2 全护线管。

AP_SEC_GUARDED_CONDUIT
护护线管；；物理T }。

AP_SEC_ENCRYPTED
在线 7 O加 \ 。

AP_SEC_GUARDED_RADIATION
护线 7 ；；物理和辐 d T }。

cos_tg_row.minimum.prop_delay
传%延Y D最小极限， m >一 v 信号在 4 7 中传M D 1 间， %位为微 k 。此值 ` k 为一 v 1 字Z D ！ c }， I 公= 0.1mmm * 2 eeeee m >，其中字Z D位 m >为 eeeeemmm。缺！值如下 y >:

AP_PROP_DELAY_MINIMUM
无传%延Y。

AP_PROP_DELAY_LAN
小Z 480 微k延Y。

AP_PROP_DELAY_TELEPHONE
在 480 和 49 512 微k间延Y。

AP_PROP_DELAY_PKT_SWITCHED_NET
在 49 512 和 245 760 微k间延Y。

AP_PROP_DELAY_SATELLITE
大Z 245 760 微k延Y。

AP_PROP_DELAY_MAXIMUM
最大传%延Y。

cos_tg_row.minimum.modem_class
t。此字段& <终h置为c。

cos_tg_row.minimum.user_def_parm_1
在 0--255 间D C户定义N } D最小极限。

cos_tg_row.minimum.user_def_parm_2
在 0--255 间D C户定义N } D最小极限。

cos_tg_row.minimum.user_def_parm_3
在 0--255 间D C户定义N } D最小极限。

cos_tg_row.maximum.effect_cap
P效能& 5际%元D最大限制。此值 ` k为一v 1 字Z D ！ c }， I公= 0.1mmm * 2 eeeee m >，其中字Z D位m >为 eeeeemmm。P效能&D ？ v %元H Z 300 位／k。

cos_tg_row.maximum.connect_cost
？，S 1间D最大限制。P效值G 0--255 之间D { }值，其中 0 G最M？，S 1间I >，而 255 G最_ ？，S 1间I >。

cos_tg_row.maximum.byte_cost
？字Z D最大限制。P效值G 0--255 之间D { }值，其中 0 G最M？字Z I >，而 255 G最_ ？字Z I >。

cos_tg_row.maximum.security
2 全性值D最大极限如下y >:

AP_SEC_NONSECURE
无2全性。

AP_SEC_PUBLIC_SWITCHED_NETWORK
在此，S网g O发M D }] +在公C ；换网g O w动。

AP_SEC_UNDERGROUND_CABLE
}]在2全S X g缆中发M。

AP_SEC_SECURE_CONDUIT
线7 G ；加#护D 2全护线管。

AP_SEC_GUARDED_CONDUIT
#护护线管；；物理T }。

AP_SEC_ENCRYPTED
在线 7 O加 \ 。

AP_SEC_GUARDED_RADIATION
护线 7 ； ；物理和辐 d T } 。

cos_tg_row.maximum.prop_delay
传%延Y D最大极限， m > 一 v 信号在 4 7 中传M D 1 间， %位为微 k 。此值 ` k 为一 v 1 字Z D ！ c } ， I 公 = 0.1mmm * 2 eeeee m > ， 其中字Z D位 m > 为 eeeeemmm。缺 ！值如下 y > :

AP_PROP_DELAY_MINIMUM
无传%延Y。

AP_PROP_DELAY_LAN
小Z 480 微 k 延Y。

AP_PROP_DELAY_TELEPHONE
在 480 和 49 512 微 k 间延Y。

AP_PROP_DELAY_PKT_SWITCHED_NET
在 49 512 和 245 760 微 k 间延Y。

AP_PROP_DELAY_SATELLITE
大Z 245 760 微 k 延Y。

AP_PROP_DELAY_MAXIMUM
最大传%延Y。

cos_tg_row.maximum.modem_class
t 。此字段& < 终 h 置为 c 。

cos_tg_row.maximum.user_def_parm_1
在 0--255 间D C 户定义N } D最大极限。

cos_tg_row.maximum.user_def_parm_2
在 0--255 间D C 户定义N } D最大极限。

cos_tg_row.maximum.user_def_parm_3
在 0--255 间D C 户定义N } D最大极限。

cos_tg_row.weight
k 此 TG 行关 * D权值。

返回N数

如果动词 I 功执行， 则L 序返回下 P N } :

primary_rc
AP_OK

如果因N } 错误而 9 动词 ; 能执行， 则L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_COS_NAME

AP_INVALID_NUMBER_OF_NODE_ROWS
AP_INVALID_NUMBER_OF_TG_ROWS
AP_NODE_ROW_WGT_LESS_THAN_LAST
AP_TG_ROW_WGT_LESS_THAN_LAST

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_COS_TABLE_FULL

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_DEFAULTS

DEFINE_DEFAULTS 允许C户定义或重新定义Z c D缺！Y作。

VCB a构

```
typedef struct define_defaults
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        DEFAULT_CHARS   default_chars;  /* default information   */
} DEFINE_DEFAULTS;

typedef struct default_chars
{
        unsigned char   description[RD_LEN];
                                        /* resource description  */
        unsigned char   mode_name[8];   /* default mode name     */
        unsigned char   implicit_plu_forbidden;
                                        /* disallow implicit     */
                                        /*  PLUs?                */
        unsigned char   specific_security_codes;
                                        /* generiuc security     */
                                        /* sense codes           */
        unsigned short  limited_timeout;/* timeout for limited   */
                                        /* sessions              */
        unsigned char   reserv[244];    /* reserved              */
} DEFAULT_CHARS;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_DEFAULTS

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

default_chars.description

资源5 w（返回= QUERY_DEFAULTS O）。|G一v在>X I显>字符集中D一v 16 字Z字符串。y P 16 v字Z都P效。

default_chars.mode_name

+作为缺！值9 C D方=D{ F。b G一v 8 字Z字母}字D A 型 EBCDIC字符串（以一v字母*头），"以 EBCDIC U q R n d。

default_chars.implicit_plu_forbidden

X制L序对未知伙i LU G否I取隐=定义（AP_YES 或 AP_NO）。

default_chars.specific_security_codes

X制L序在2全性认证或Z权' \ 1 G否9 C X定检b k （AP_YES 或AP_NO）。注意，X定检b k + v返回x某些伙i LU，那些伙i LU 已在会话O(f \支V。

default_chars.limited_timeout
指定，1 值，在 C ，1 值后 U 闲限制资源： y $ _ 会话 + ；M放。范围在 0 = 65535 k 间。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

如果动词指定 D 缺！方 = 无效（例如，非 EBCDIC A），或 _ 缺！方 = P 效 + P 未定义， L 序 + 返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_MODE_NAME

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

？ v 字段 D 重定义 D 效果如下:

description
重定义 " 即 z 效。 | 新 5 w 返回 = 后继 D QUERY_DEFAULT 动词 O。

mode_name
一 v 重定义 D 效果 & C Z y P 后继 D 隐 = 方 = 定义，例如， | 新方 = + 作为缺！方 = 9 C。在一 v 以 0 未知方 = 中 D 某一重定义 D 效果（例如，继 P 以 0 D 缺！方 = X 性）H 同 Z 未知方 = D 一 v 重定义。例如，如果缺！方 = 为 #INTER， R L 序对 Z（未知 D ）MODE1 S U 一 v bIND，则在以后 + MODE1 D 缺！方 = 重定义为 #BATCH D 效果， k 9 C DEFINE_MODE(MODE1) + 方 = X 性指定为 #BATCH D 效果 & C 一致。

implicit_plu_forbidden
重定义 " 即 z 效。 y P 后继隐 = PLU 定义 I 功 k 否取 v Z 此字段 D | 新值。

specific_security_codes
重定义 " 即 z 效。

limited_timeout
| 新 D 值 C Z y P 在重定义后 (" D 新会话。 I 值 C Z 现 P D 会话。

DEFINE_DEFAULT_PU

DEFINE_DEFAULT_PU 允许C户定义、重定义或修D缺！ PU D任何字段。 | 还允许C户> } 缺！ PU，方法G指定一v U PU { F。如果未在一v TRANSFER_MS_DATA 动词O显=指定 PU { F，则在 TRANSFER_MS_DATA 动词O D管理服务信息+发M=缺！ PU D会话，主机为 SSCP。如需 | 多信息， k N阅 Z603页D 『 Z15B 管理服务动词』。

VCB a构

```
typedef struct define_default_pu
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   pu_name[8];     /* PU name               */
        unsigned char   description[RD_LEN];
                                        /* resource description  */
        unsigned char   reserv3[16];    /* reserved              */
} DEFINE_DEFAULT_PU;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_DEFAULT_PU

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

pu_name

+作为缺！值9 C D >X PU D { F。b G一v 8 字Z字母}字D A 型EBCDIC 字符串（以一v字母*头）， " 以 EBCDIC U q R n d。

description

资源5 w（返回= QUERY_DEFAULT_PU O）。 | G一v在>X I 显>字符集中D一v 16 字Z字符串。y P 16 v字Z都P效。

返回N数

如果动词I 功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_DLC

DEFINE_DLC 定义一 v 新D DLC 或修D一 v 现P D DLC。此动词定义 DLC { F（在{ v Z c中唯一）和一些 DLC-X定 }]（k基> a构" 置在一起）。此 }] 在 DLC u <化 1 9 C，" R其q = X定Z DLC 类型（例如n牌环）。只P = 加= 动词D DLC-X定 }] E I 9 C DEFINE_DLC 动词x 行修D。要b 样做， X须W先发v 一 v STOP_DLC 动词，以9 DLC 处Z 4位状，。

如需P关 DLC、端Z和4 7 >间关系D | 多信息， k N阅Z 16页D『DLC 过L、端Z和4 7 >』。

VCB a 构

```
typedef struct define_dlc
{
        unsigned short  opcode;            /* verb operation code         */
        unsigned char   attributes;        /* verb attributes             */
        unsigned char   reserv2;           /* reserved                    */
        unsigned char   format;            /* format                      */
        unsigned short  primary_rc;        /* primary return code         */
        unsigned long   secondary_rc;      /* secondary return code       */
        unsigned char   dlc_name[8];       /* name of DLC                 */
        DLC_DEF_DATA    def_data;          /* DLC defined data            */
} DEFINE_DLC;

typedef struct dlc_def_data
{
        DESCRIPTION     description;       /* resource description        */
        unsigned char   dlc_type;          /* DLC type                    */
        unsigned char   neg_ls_supp;       /* negotiable LS support       */
        unsigned char   port_types;        /* allowable port types        */
        unsigned char   hpr_only;          /* DLC only supports HPR links:*/
        unsigned char   reserv3;           /* reserved                    */
        unsigned char   retry_flags;       /* conditions for automatic    */
                                           /* retries                     */
        unsigned short  max_activation_attempts;
                                           /* how many automatic retries? */
        unsigned short  activation_delay_timer;
                                           /* delay between automatic     */
                                           /* retries                     */
        unsigned char   reserv4[4];           /* reserved                 */
        unsigned short  dlc_spec_data_len; /* Length of DLC specific data */
} DLC_DEF_DATA;
```

提供N数

& C L序a 供下P N }:

opcode

AP_DEFINE_DLC

attributes

动词D t 性。此字段G一 v 位字段。 Z 一位 | 含要定义资源D I 见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q = 。 + 此字段 h 置为 c，以指定在O f P v VCB D f > 。

dlc_name

DLC D { F 。 | G 一 v 在 > X I 显 > 字符集中D 一 v 8 字 Z 字符串。 y P 8 v 字 Z 都G P 效D， " R X 须 h 置。对Z OEM h 8，此 { F G X 定 Z 制造 L D 。 P 效值为 LAN、SDLC、AnyNet、X25 或 TWINAX （C U q n d = 8 v 字符)。

def_data.description

资源 5 w（返回 = QUERY_QUERY_DLC O)。 | G 一 v 在 > X I 显 > 字符集中D 一 v 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

def_data.dlc_type

DLC D类型。 v 人通信和通信服务器支 V 下 P 类型:

AP_ANYNET
AP_LLC2
AP_OEM_DLC
AP_SDLC
AP_TWINAX
AP_X25

def_data.neg_ls_supp

指定 DLC G否支 V I 协 L 4 7 >（AP_YES 或 AP_NO)。如果 **dlc_type** 为 AP_TWINAX，则 v 支 V AP_NO。如果 **dlc_type** 为 AP_ANYNET，则 v 支 V AP_YES。

def_data.port_types

指定 y a 供 **dlc_type** D 允许端 Z 类型。其值对 & Z 下 P 值 OR（或 Y 作）组合中D 一 v 或多 v 。

AP_PORT_NONSWITCHED
AP_PORT_SWITCHED
AP_PORT_SATF

9 C 下 m 来 h 置相 & DLC 类型 D 字段。

m 2. DLC 类型 D 端 Z 类型

DLC ` M	**端口** ` M
AP_ANYNET	AP_PORT_SATF
AP_LLC2	AP_PORT_SATF
AP_OEM_DLC	AP_PORT_SWITCHED 或 AP_PORT_NONSWITCHED
AP_SDLC	AP_PORT_SWITCHED 或 AP_PORT_NONSWITCHED
AP_TWINAX	AP_PORT_NONSWITCHED
AP_X25	AP_PORT_SWITCHED 或 AP_PORT_NONSWITCHED

def_data.max_activation_attempts

此字段指定 DLC G否 v 支 V HPR 4 7 。对 Z IP 4 7 O D HPR， C 值 X 须 h 置为 AP_YES。

AP_YES
AP_NO

def_data.retry_flags

此字段指定４７＞Ｉ自动重ＴＤｕ件。｜Ｇ一ｖ位字段，＂Ｉ９Ｃ任何下Ｐ４位 OR（或Ｙ作）组合。

AP_RETRY_ON_START

在＂Ｔ激活１，如果未从远ＬＺｃＳＵ＝响&，则重Ｔ４７激活。如果在＂Ｔ激活１，基＞端ＺＧ非活动Ｄ，则Ｌ序＋＂Ｔ激活｜。

AP_RETRY_ON_FAILURE

如果在活动和暂挂活动１４Ｓ＇＼，则重Ｔ４７激活。如果在＂Ｔ激活１，基＞端Ｚ发ｚ故Ｏ，则Ｌ序＋＂Ｔ激活｜。

AP_RETRY_ON_DISCONNECT

如果４７Ｉ远ＬＺｃ以｝＃方＝停止，则重Ｔ４７激活。

AP_DELAY_APPLICATION_RETRIES

Ｉ＆ＣＬ序启动Ｄ４７激活重Ｔ（９Ｃ START_LS 或ｋｓ＝４７激活）＋９Ｃ **activation_delay_timer** ｘ行ｗ＝。

AP_INHERIT_RETRY

此ｊ志无效果。

def_data.max_activation_attempts

此字段；ｚｚ效果，｝非在 **def_data.retry_flags** 中Ｄ DEFINE_LS 中至Ｙｈ置一ｖｊ志，DEFINE_LS ＯＤ **def_data.max_activation_attempts** ｈ置为 AP_USE_DEFAULTS，以及 DEFINE_PORT ＯＤ **def_data.max_activation_attempts** ｈ置为 AP_USE_DEFAULTS。

此字段指定１远ＬＺｃ无响&或基＞端Ｚ；活动１，Ｌ序允许重ＴＤ次｝。｜｜括自动重Ｔ和Ｉ＆ＣＬ序｝动Ｄ激活＂Ｔ＝＿。

如果达＝此极限，则；再ｘ行自动重Ｔ。Ｃｕ件Ｉ STOP_LS、STOP_PORT、STOP_DLC 或一ｖＩ功激活４位。START_LS 或 OPEN_LU_SSCP_SEC_RQ ｚｚ一ｖ％ｖ激活＂Ｔ，｜在激活＇＼１；再重Ｔ。

ｃｍ＞ ’无限制’。值 AP_USE_DEFAULTS ｍ＞ ’无限制’。

def_data.activation_delay_timer

此字段；ｚｚ效果，｝非在 **def_data.retry_flags** 中Ｄ DEFINE_LS 中至Ｙｈ置一ｖｊ志，DEFINE_LS ＯＤ **def_data.max_activation_attempts** ｈ置为 AP_USE_DEFAULTS，以及 DEFINE_PORT ＯＤ **def_data.max_activation_attempts** ｈ置为 AP_USE_DEFAULTS。

此字段指定在自动重Ｔ间Ｌ序Ｈ待Ｄｋ｝，以及如果在 **def_data.retry_flags** 中ｈ置 AP_DELAY_APPLICATION_RETRIES 位，Ｉ＆ＣＬ序｝动Ｄ激活＂Ｔ间Ｄｋ｝。

值ｃ或 AP_USE_DEFAULTS ９Ｃ缺！计１器值 30 ｋ。

def_data.dlc_spec_data_len

此字段&＜终ｈ置为ｃ。

返回N数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DLC_NAME

AP_INVALID_DLC_TYPE
AP_INVALID_RETRY_FLAGS
AP_INVALID_PORT_TYPE
AP_HPR_NOT_SUPPORTED

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_DLC_ACTIVE

AP_INVALID_DLC_TYPE
AP_CANT_MODIFY_VISIBILITY

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_DLUR_DEFAULTS

DEFINE_DLUR_DEFAULTS 允许C户定义、重定义或取消一v缺！关* LU 服务器（DLUS）和一v 8份缺！ DLUS。 1 DLUS 启动某些 PU D SSCP-PU 激活，而b些 PU ；P一v w确指定D关* DLUS 1， DLUS + 9 C缺！ DLUS { F。如果在DEFINE_DLUR_DEFAULTS 动词O未w确指定 DLUS { F，则1 0缺！（或8份DLUS）+ ；取消。

VCB a构

```
typedef struct define_dlur_defaults
{
        unsigned short  opcode;             /* verb operation code   */
        unsigned char   reserv2;            /* reserved              */
        unsigned char   format;             /* format                */
        unsigned short  primary_rc;         /* primary return code   */
        unsigned long   secondary_rc;       /* secondary return code */
        unsigned char   description[RD_LEN];
                                            /* resource description  */
        unsigned char   dlus_name[17];      /* DLUS name             */
        unsigned char   bkup_dlus_name[17]; /* Backup DLUS name      */
        unsigned char   reserv3;            /* reserved              */
        unsigned short  dlus_retry_timeout; /* DLUS Retry Timeout    */
        unsigned short  dlus_retry_limit;   /* DLUS Retry Limit      */
        unsigned char   reserv4[16];        /* reserved              */
} DEFINE_DLUR_DEFAULTS;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_DLUR_DEFAULTS

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

description

资源5 w。 | G一v在> X I显>字符集中D一v 16 字Z字符串。y P 16 v字Z都P效。

dlus_name

+作为缺！值9 C D DLUS Z c D { F。C值& h置为全c，或I = v A 型EBCDIC 字符串组I、中间以一v EBCDIC c ，S、"以 EBCDIC U q R n d D 17 字Z字符串。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）如果此字段h置为全c，则1 0缺！ DLUS + ；取消。

bkup_dlus_name

+作为8份缺！值9 C D DLUS Z c D { F。C值& h置为全c，或I = v A型 EBCDIC 字符串组I、中间以一v EBCDIC c ，S、"以 EBCDIC U q R n d D 17 字Z字符串。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）如果此字段h置为全c，则1 0 8份缺！ DLUS + ；取消。

dlus_retry_timeout

在Z二次和后继" T * 系 DLUS 间D间t （ k）。u < " T和Z一次重T之间D间t <终为 1 k。如果指定c，则+ 9 C缺！值 5 k。

dlus_retry_limit
在Z一次 ' \后继续 * 系 DLUS D最大重T次 } 。如果指定c , 则+ 9 C缺 ! 值 3 k。如果指定 X'FFFF', v人通信或通信服务器 + ；确定X x行重T。

返回N数

如果动词I功执行, 则L序返回下P N } :

primary_rc
AP_OK

如果因N } 错误而9动词；能执行, 则L序返回下P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DLUS_NAME

AP_INVALID_BKUP_DLUS_NAME

如果因Z c P未启动而9动词；能执行, 则L序返回下P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行, 则L序返回下P N } :

primary_rc
AP_NODE_STOPPING

如果因系统错误而9动词；能执行, 则L序返回下P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_DOWNSTREAM_LU

此动词ｖＪＣＺ通信服务器。

DEFINE_DOWNSTREAM_LU ＣＺ ＰＵ 集中。１９Ｃ ＰＵ 集中１，下Ｎ ＬＵ Ｉｋ ＯＮ 主机ｘ行通信。要ｂ样做，通信服务器＋？ｖ下Ｎ ＬＵ ３ｄ＝一ｖ关＊＞Ｘ ＬＵ，引Ｃ为 *主机 LU*。

DEFINE_DOWNSTREAM_LU 定义一ｖ新Ｄ下Ｎ ＬＵ，＂Ｒ；能ＣＺ修Ｄ现ＰＤ定义。下Ｎ ＬＵ ３ｄ＝指定Ｄ主机 ＬＵ （９Ｃ DEFINE_LU_0_TO_3 动词定义）。主机 ＬＵ 也Ｉｙ］ ＬＵ Ｘ指定。

１ DEFINE_DOWNSTREAM_LU 为一ｖ现ＰＤ下Ｎ ＬＵ 定义发ｖ１，则定义Ｘ须ＧＨ同Ｄ。如果下Ｎ４７活动Ｒ下Ｎ ＬＵ 非活动，则动词＋Ｉ功返回，＂Ｒ执行一ｖ重激活＂Ｔ（！管Ｉ能；Ｉ功）。如果下Ｎ非活动或下Ｎ ＬＵ 已激活，则动词＋＇＼。重激活＂ＴＤ处理取ｖＺｙ指定主机 ＬＵ Ｄ状，。

- 如果主机 ＬＵ 活动，则 ACTLU ＋＂即重发Ｍ＝下Ｎ ＬＵ。
- 如果主机 ＬＵ 非活动，则Ｚｃ在发Ｍ ACTLU ＝下Ｎ ＬＵ ０＋Ｈ待主机 ＬＵ ；激活。如果至主机Ｄ４７非活动，则Ｚｃ＋＂Ｔ激活｜（如果主机４７；能自动；激活，ｂ＋；会Ｉ功）。

VCB a构

```
typedef struct define_downstream_lu
{
        unsigned short  opcode;           /* verb operation code        */
        unsigned char   attributes;       /* verb attributes            */
        unsigned char   reserv2;          /* reserved                   */
        unsigned char   format;           /* format                     */
        unsigned short  primary_rc;       /* primary return code        */
        unsigned long   secondary_rc;     /* secondary return code      */
        unsigned char   dslu_name[8];     /* Downstream LU name         */
        DOWNSTREAM_LU_DEF_DATA def_data;  /* defined data               */
} DEFINE_DOWNSTREAM_LU;

typedef struct downstream_lu_def_data
{
        unsigned char   description[RD_LEN];
                                          /* resource description       */
        unsigned char   nau_address;      /* Downstream LU NAU address  */
        unsigned char   dspu_name[8];     /* Downstream PU name         */
        unsigned char   host_lu_name[8];  /* Host LU or Pool name       */
        unsigned char   allow_timeout;    /* Allow timeout of host LU?  */
        unsigned char   delayed_logon;    /* Allow delayed logon to     */
                                          /* host LU                    */
        unsigned char   reserv2[6];       /* reserved                   */
} DOWNSTREAM_LU_DEF_DATA;
```

提供Ｎ数

＆ＣＬ序ａ供下ＰＮ｝：

opcode

AP_DEFINE_DOWNSTREAM_LU

DEFINE_DOWNSTREAM_LU

attributes

动词D t 性。此字段G一v 位字段。 Z 一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

dslu_name

；定义下N LU D{ F。 b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）， "以 EBCDIC U q R n d。

def_data.description

资源5 w（返回= QUERY_DOWNSTREAM_LU O）。此字段D $度&为 4 v字Z D 6 }， R ；为c。

def_data.nau_address

DOWNSTREAM LU D网g I 寻址%元X址。值X须在 1-255 间。

def_data.dspu_name

DOWNSTREAM PU（在 DEFINE_LS 中指定）D{ F。 b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）， "以 EBCDIC U q R n d。

def_data.host_lu_name

下N LU 3 d = D主机 LU 或主机 LU X D{ F。 b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）， "以 EBCDIC U q R n d。

def_data.allow_timeout

指定如果会话；活动D 1 间达=在主机 LU 定义中指定D超时值， G否允许L序对此下N LU y 9 C D主机 LU ，1（AP_YES 或 AP_NO）。

def_data.delayed_logon

指定L序G否&延Y下N LU k主机 LU D ，S，直至S U = Z 一v来自下N LU D }]。相反，发M一v模拟注a屏幕=下N LU（AP_YES 或 AP_NO）。

返回N数

如果动词I 功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因N }错误而9 动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_DNST_LU_NAME

AP_INVALID_NAU_ADDRESS

如果因状，错误而9 动词；能执行，则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_PU_NAME

AP_INVALID_PU_TYPE
AP_PU_NOT_DEFINED
AP_LU_ALREADY_DEFINED
AP_LU_NAU_ADDR_ALREADY_DEFD
AP_INVALID_HOST_LU_NAME
AP_LU_NAME_POOL_NAME_CLASH
AP_PU_NOT_ACTIVE
AP_LU_ALREADY_ACTIVATING
AP_LU_DEACTIVATING
AP_LU_ALREADY_ACTIVE
AP_CANT_MODIFY_VISIBILITY
AP_INVALID_ALLOW_TIMEOUT
AP_INVALID_DELAYED_LOGON
AP_DELAYED_VERB_PENDING

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_DOWNSTREAM_LU_RANGE

此动词ｖＪＣＺ通信服务器。

DEFINE_DOWNSTREAM_LU_RANGE ＣＺ PU 集中。１９Ｃ PU 集中１，下Ｎ LU ＩｋＯＮ主机ｘ行通信。要ｂ样做，通信服务器＋？ｖ下Ｎ LU ３ｄ＝一ｖ关＊＞Ｘ LU，引Ｃ为 *主机 LU*。

DEFINE_DOWNSTREAM_LU_RANGE 允许在一ｖ指定Ｄ NAU 范围内定义多ｖ下Ｎ LU （＋；ＩＣ来修Ｄ现ＰＤ定义）。ＺｃＹ作员ａ供一ｖ基｛和一ｖ NAU 范围。LU ｛ＦＩ＋基｛和 NAU Ｘ址组合在一起ｚＩ。

例如，一ｖ基｛ LUNME ｋ NAU 范围 1-4 组合在一起，＋定义 LU LUNME001、LUNME002、 LUNME003 和 LUNME004。 一ｖ基｛Ｄ＄度小Ｚ ５ ｖ非ｎｄ字符＋９Ｃ LU ｛ＦＤ＄度小Ｚ ８ ｖ字符。ｂ１，通信服务器＋ｘ行Ｒｎｄ以达＝ ８ ｖ字符。

？ｖ下Ｎ LU ３ｄ＝指定Ｄ主机 LU （９Ｃ DEFINE_LU_0_TO_3 动词定义）。

VCB ａ构

```
typedef struct define_downstream_lu_range
{
        unsigned short  opcode;           /* verb operation code        */
        unsigned char   attributes;       /* verb attributes            */
        unsigned char   reserv2;          /* reserved                   */
        unsigned char   format;           /* format                     */
        unsigned short  primary_rc;       /* primary return code        */
        unsigned long   secondary_rc;     /* secondary return code      */
        unsigned char   dslu_base_name[5];/* Downstream LU base name    */
        unsigned char   description[RD_LEN];
                                          /* resource description       */
        unsigned char   min_nau;          /* min NAU address in range   */
        unsigned char   max_nau;          /* max NAU address in range   */
        unsigned char   dspu_name[8];     /* Downstream PU name         */
        unsigned char   host_lu_name[8];  /* Host LU or pool name       */
        unsigned char   allow_timeout;    /* Allow timeout of host LU?  */
        unsigned char   delayed_logon;    /* Allow delayed logon to the */
                                          /* host LU                    */
        unsigned char   reserv4[6];       /* reserved                   */
} DEFINE_DOWNSTREAM_LU_RANGE;
```

提供Ｎ数

＆ＣＬ序ａ供下ＰＮ｝:

opcode

AP_DEFINE_DOWNSTREAM_LU_RANGE

attributes

动词Ｄｔ性。此字段Ｇ一ｖ位字段。Ｚ一位｜含要定义资源ＤＩ见性，＂ｋ下Ｐ之一对＆:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q = 。 + 此字段 h 置为 c，以指定在 O f P v VCB D f > 。

dslu_base_name

下 N LU { F 范围 D 基 { 。 b G 一 v 5 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。此基 { 之后 = 加 3 v A 型 EBCDIC } 字字符， m > 在 NAU 范围内 ? v LU NAU X 址 D . x 制值。

description

资源 5 w（返回 = QUERY_DOWNSTREAM_LU O）。此字段 D $ 度 & 为 4 v 字 Z D 6 }， R ；为 c 。

min_nau

范围内 D 最小 NAU X 址。 | I 为 1 至 255（ | 括 1 和 255）。

max_nau

范围内 D 最大 NAU X 址。 | I 为 1 至 255（ | 括 1 和 255）。

dspu_name

DOWNSTREAM PU（在 DEFINE_LS 中指定）D { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。

host_lu_name

y P 在范围内 D 下 N LU 3 d = D 主机 LU 或主机 LUs X D { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。

allow_timeout

指定如果会话；活动 D 1 间达 = 在主机 LU 定义中指定 D **超时**值， G 否允许 L 序对此下 N LU y 9 C D 主机 LU ，1（AP_YES 或 AP_NO）。

delayed_logon

指定 L 序 G 否 & 延 Y 下 N LU k 主机 LU D ， S ，直至 S U = Z 一 v 来自下 N LU D }] 。相反， + 发 M 一 v 模拟注 a 屏幕 = 下 N LU（AP_YES 或 AP_NO）。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N }：

primary_rc

AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_DNST_LU_NAME

AP_INVALID_NAU_ADDRESS
AP_INVALID_ALLOW_TIMEOUT
AP_INVALID_DELAYED_LOGON

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_LU_NAME_POOL_NAME_CLASH

AP_LU_ALREADY_DEFINED
AP_INVALID_HOST_LU_NAME
AP_PU_NOT_DEFINED
AP_INVALID_PU_NAME
AP_INVALID_PU_TYPE
AP_LU_NAU_ADDR_ALREADY_DEFD
AP_CANT_MODIFY_VISIBILITY
AP_DELAYED_VERB_PENDING

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_DSPU_TEMPLATE

此动词ｖＪＣＺ通信服务器。

此动词ＣＺ ＰＵ 集中。 １９Ｃ ＰＵ 集中１，下Ｎ ＬＵ Ｉｋ ＯＮ主机ｘ行通信。要ｂ样做，通信服务器＋？ｖ下Ｎ ＬＵ ３ｄ＝一ｖ关＊＞Ｘ ＬＵ，引Ｃ为主机LU。DEFINE_DSPU_TEMPLATE 为在下Ｎ工作＞组中Ｄ下Ｎ LU 定义一ｖ模ｅ。 １一ｖ工作＞在某一隐＝４７（以０未定义）Ｏ，Ｓ＝通信服务器，则此模ｅ；ＣＺ代ｆ下Ｎ LU 定义Ｄ位置。ｂ些模ｅＩ DEFINE_PORT 动词ＯＤ **implicit_dspu_template** 字段引Ｃ。DEFINE_DSPU_TEMPLATE 既ＩＣＺ定义新Ｄ模ｅ，也ＩＣＺ修Ｄ现ＰＤ模ｅ（ｄ然ｙ修Ｄ模ｅＤ现Ｐ５例；＼０响）。

VCB ａ构

```
typedef struct define_dspu_template
{
        unsigned short  opcode;              /* verb operation code       */
        unsigned char   attributes;          /* verb attributes           */
        unsigned char   format;              /* format                    */
        unsigned short  primary_rc;          /* primary return code       */
        unsigned long   secondary_rc;        /* secondary return code     */
        unsigned char   template_name[8];    /* name of template          */
        unsigned char   description;         /* resource description      */
        unsigned char   modify_template;     /* Modify existing template? */
        unsigned char   reserv1[11];         /* reserved                  */
        unsigned short  max_instance;        /* Max active template       */
                                             /* instances                 */
        unsigned short  num_of_dslu_templates;
                                             /* number of DSLU templates  */
} DEFINE_DSPU_TEMPLATE;

typedef struct dslu_template
{
        unsigned char   min_nau;             /* min NAU address in range  */
        unsigned char   max_nau;             /* max NAU address in range  */
        unsigned char   allow_timeout;       /* Allow timeout of host LU? */
        unsigned char   delayed_logon;       /* Allow delayed logon to    */
                                             /* host LU                   */
        unsigned char   reserv1[8];          /* reserved                  */
        unsigned char   host_lu[8];          /* host LU or pool name      */
} DSLU_TEMPLATE;
```

提供Ｎ数

＆ＣＬ序ａ供下ＰＮ｝:

opcode

AP_DEFINE_DSPU_TEMPLATE

attributes

动词Ｄｔ性。此字段Ｇ一ｖ位字段。Ｚ一位｜含要定义资源ＤＩ见性，＂ｋ下Ｐ之一对＆:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

DEFINE_DSPU_TEMPLATE

format

j 6 VCB D q = 。 + 此字段 h 置为 c，以指定在O f P v VCB D f > 。

template_name

DSPU 模 e D { F 。(| k 在 PORT_DEF_DATA D **implicit_dspu_template** 字段中指定D { F 相对&)。 | G 一 v 在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须 h 置。

description

资源 5 w（返回 = QUERY_DSPU_TEMPLATE O)。 | D $ 度& 为 4 v字Z D 6 }， R ；为 c 。

modify_template

指定此动词G否& m加 = 加D DSLU 模 e = 现P DSPU 模 e 中，或 _ & f 换现 P D DSPU （AP_MODIFY_DSPU_TEMPLATE 或 AP_REPLACE_DSPU_TEMPLATE)。

如果 modify_template ； h 置为 AP_MODIFY_DSPU_TEMPLATE，而 _ { DSPU 模 e ；存在，则 + 创 (DSPU 模 e 。

如果 **modify_template** ； h 置为 AP_MODIFY_DSPU_TEMPLATE，而 _ { DSPU 模 e ；存在，则 = 加D DSPU 模 e + ；m加 = 现 P D DSPU 模 e 中。

如果 **modify_template** ； h 置为 AP_REPLACE_DSPU_TEMPLATE，则 + 创 (一 v 新D模 e 。 | I 为 0 至 65535（ | 括 0 和 65535)，其中 0 m > 无限制。

max_instance

b G同 1 I 激活模 e 5 例D最大 } 目。 1 达 = 此极限 1， ；能再创 (新D 5 例。 | I 为 0 至 65535（ | 括 0 和 65535)，其中 0 m > 无限制。

num_of_dslu_templates

z 在 DEFINE_DSPU_TEMPLATE VCB 之后D DSLU 模 e 2 G D } 目。 | I 为 0 至 255（ | 括 0 和 255)。

dslu_template.min_nau

范围内D最小 NAU X址。 | I 为 1 至 255（ | 括 1 和 255)。

dslu_template.max_nau

范围内D最大 NAU X址。 | I 为 1 至 255（ | 括 1 和 255)。

def_data.allow_timeout

指定如果会话；活动D 1 间达 = 在主机 LU 定义中指定D**超时**值，G否允许L序对此下N LU y 9 C D主机 LU ，1（AP_YES 或 AP_NO)。

def_data.delayed_logon

指定L序G否&延Y下N LU k 主机 LU D ，S，直至S U = Z 一 v 来自下N LU D }] 。相反，发M一 v 模拟注 a 屏幕 = 下N LU（AP_YES 或 AP_NO)。

dslu_template.host_lu

y P 在范围内D下N LU + 3 d = D主机 LU 或主机 LUs X D { F 。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头)，" 以 EBCDIC U q R n d 。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc
AP_OK

如果因N}错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_TEMPLATE_NAME

AP_INVALID_NAU_ADDRESS
AP_INVALID_NAU_RANGE
AP_CLASHING_NAU_RANGE
AP_INVALID_NUM_DSPU_TEMPLATES
AP_INVALID_ALLOW_TIMEOUT
AP_INVALID_DELAYED_LOGON
AP_INVALID_MODIFY_TEMPLATE

如果因状，错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_INVALID_HOST_LU_NAME

AP_CANT_MODIFY_VISIBILITY

如果因相关 START_NODE N}未h置而9动词；能执行，则L序返回下PN}:

primary_rc
AP_FUNCTION_NOT_SUPPORTED

如果因Z c P未启动而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_FOCAL_POINT

v人通信或通信服务器ＩＰｋ；同９ｃＤ多种类型Ｄ关系。DEFINE_FOCAL_POINT 定义一ｖ９ｃ， ｖ人通信或通信服务器ｋ此９ｃ＿Ｐ隐＝关系（类型Ｉ为主９ｃ或８份９ｃ）。ｂ些关系及其（１．９#2为X8Kc0 ＴＤ（ｗ）Ｔｊ／Ｆ２１ １ Ｔｆ１．０２２２ ０

VCB a 构

```
typedef struct define_focal_point
{
        unsigned short  opcode;           /* verb operation code          */
        unsigned char   reserv2;          /* reserved                     */
        unsigned char   format;           /* format                       */
        unsigned short  primary_rc;       /* primary return code          */
        unsigned long   secondary_rc;     /* secondary return code        */
        unsigned char   reserved;         /* reserved                     */
        unsigned char   ms_category[8];   /* management services category */
        unsigned char   fp_fqcp_name[17]; /* Fully qualified focal        */
                                          /* point CP name                */
        unsigned char   ms_appl_name[8];  /* Focal point application name */
        unsigned char   description[RD_LEN];
                                          /* resource description         */
        unsigned char   backup;           /* is focal point a backup      */
        unsigned char   reserv3[16];      /* reserved                     */
} DEFINE_FOCAL_POINT;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_DEFINE_FOCAL_POINT

format

j 6 VCB D q =。+此字段 h 置为 c，以指定在O f P v VCB D f >。

ms_category

管理服务类 p。| I 为如在 SNA 管理服务中定义D管理服务类 p D、以 4 字 Z a 构定义D值（以 EBCDIC U q R n d）之一，或为一 v 8 字 Z 1134 EBCDIC 2 装定义类型D { F。

fp_fqcp_name

9 c D全限定X制 c { F。C值& h 置为全 c，或 I = v A 型 EBCDIC 字符串组 I、中间以一 v EBCDIC c ，S、 " 以 EBCDIC U q R n d D 17 字 Z 字符串。（？v { F D $ 度最多 I 为 8 v字Z， " ；含 6 入U q。） 如果 9 c ；取消，则此字段& h 置为全 c。

ms_appl_name

9 c & C L 序{ F。| I 为如在 SNA 管理服务中定义D管理服务& C L 序D、以 4 字 Z a 构定义D值（以 EBCDIC U q R n d）之一，或为一 v 8 字 Z 1134 EBCDIC 2 装定义类型D { F。如果 9 c ；取消，则此字段& h 置为全 c。

description

资源 5 w（返回 = QUERY_DEFAULTS_POINT O）。| G 一 v 在 > X I 显 > 字符集中D一 v 16 字 Z 字符串。y P 16 v字 Z 都 P 效。

backup

指定G否定义一 v 8 份 9 c（AP_YES 或 AP_NO）。如果 1 0 活动D 9 c ；取消，则此字段 # t 。（议 9 C DELETE_FOCAL_POINT 动词（指定 AP_ACTIVE）来取消 1 0 活动D 9 c 。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_FP_NAME

AP_INVALID_CATEGORY_NAME

如果动词未I功执行，则L序返回下P N }:

primary_rc
AP_REPLACED

AP_UNSUCCESSFUL

secondary_rc
AP_IMPLICIT_REQUEST_REJECTED

AP_IMPLICIT_REQUEST_FAILED

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误或L序无法k 9 c I功取C * 系而9动词；能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_INTERNAL_PU

DEFINE_INTERNAL_PU 动词定义 I DLUR a 供服务 D > X PU。此动词；C Z 定义 k 主机直 S ，S D > X PU。如需定义 k 主机直 S ，S D > X PU， k N 阅 Z 76 页 D 『DEFINE_LS』。

"：& 9 C DEFINE_LS 动词定义下 P 对象:

- I 以下 a 供服务 D 下 N PU:
  - DLUR
  - PU 集中
- k 主机直 S ，S D > X PU

VCB a 构

```
typedef struct define_internal_pu
{
        unsigned short  opcode;             /* verb operation code   */
        unsigned char   attributes;         /* verb attributes       */
        unsigned char   format;             /* format                */
        unsigned short  primary_rc;         /* primary return code   */
        unsigned long   secondary_rc;       /* secondary return code */
        unsigned char   pu_name[8];         /* internal PU name      */
        INTERNAL_PU_DEF_DATA def_data;      /* defined data          */
} DEFINE_INTERNAL_PU;

typedef struct internal_pu_def_data
{
        unsigned char   description[RD_LEN];
                                            /* resource description  */
        unsigned char   dlus_name[17];      /* DLUS name             */
        unsigned char   bkup_dlus_name[17]; /* backup DLUS name      */
        unsigned char   pu_id[4];           /* PU identifier         */
        unsigned short  dlus_retry_timeout; /* DLUS retry timeout                */
        unsigned short  dlus_retry_limit;   /* DLUS retry limit                  */
        unsigned char   conventional_lu_compression;
                                            /* Data compression      */
                                            /* requested for con-    */
                                            /* ventional LU sessions */
        unsigned char   conventional_lu_cryptography;
                                            /* Cryptography required */
                                            /* for conventional LU   */
                                            /* sessions              */
        unsigned char   reserv2[2];         /* reserved           */
} INTERNAL_PU_DEF_DATA;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_DEFINE_INTERNAL_PU

attributes

动词 D t 性。此字段 G 一 v 位字段。Z 一位 | 含要定义资源 D I 见性， " k 下 P 之一对 &:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

DEFINE_INTERNAL_PU

format

j 6 VCB D q = 。 + 此字段 h 置为 c，以指定在O f P v VCB D f > 。

pu_name

y 定义内？ PU D { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d 。

def_data.description

资源 5 w（返回 = QUERY_DLUR_PU 和 QUERY_PU O）。| G 一 v 在 > X I 显 > 字符集中 D 一 v 16 字 Z 字符串。y P 16 v 字 Z 都 P 效。

def_data.dlus_name

1 DLUS 启动 SSCP-PU 激活 1， + 9 C D DLUS Z c D { F 。C 值 & h 置为全 c，或 I = v A 型 EBCDIC 字符串组 I 、中间以一 v EBCDIC c ，S 、" 以 EBCDIC U q R n d D 17 字 Z 字符串。（？v { F D $ 度最多 I 为 8 v 字 Z，" ；含 6 入 U q 。） 如果此字段 h 置为全 c，则 + 在 I DLUS 启动 D SSCP-PU 激活中 9 C 全 V 缺！ DLUS （如果已 9 C DEFINE_DLUR_DEFAULTS 动词定义）。

def_data.bkup_dlus_name

+ 作为此 PU D 8 份 DLUS 9 C D DLUS Z c D { F 。C 值 & h 置为全 c，或 I = v A 型 EBCDIC 字符串组 I 、中间以一 v EBCDIC c ，S 、" 以 EBCDIC U q R n d D 17 字 Z 字符串。（？v { F D $ 度最多 I 为 8 v 字 Z，" ；含 6 入 U q 。） 如果此字段 h 置为全 c，则全 V 8 份缺！ DLUS （如果已 9 C DEFINE_DLUR_DEFAULTS 动词定义）+ 作为此 PU D 8 份 9 C 。

def_data.pu_id

PU j 6 。| G 一 v 4 字 Z . y x 制字符串。位 0--11 h 置为 i 号，位 12--31 h 置为唯一 j 6 PU D ID 号。| X 须 k 在主机中配置 D **pu_id** 匹配。

def_data.dlus_retry_timeout

在 Z 二次和后继 " T k 在 **def_data.dlus_name** 和 **def_data.bkup_dlus_name** 字段中指定 D DLUS 取 C * 系间 D 间 t （k）。u < " T 和 Z 一次重 T 之间 D 间 t < 终为 1 k 。如果指定 c，则 + 9 C 通过 DEFINE_DLUR_DEFAULTS 配置 D 缺！值。如果 **def_data.dspu_services** 未 h 置为 AP_DLUR，则此字段 + ；忽 T 。

def_data.dlus_retry_limit

在 Z 一次 ' \ 后继续 k 在 **def_data.dlus_name** 和 **def_data.bkup_dlus_name** 字段中指定 D DLUS 取 C * 系 D 最大重 T 次 } 。如果指定 c，则 + 9 C 通过 DEFINE_DLUR_DEFAULTS 配置 D 缺！值。如果指定 X'FFFF'， L 序 + ；确定 X x 行重 T 。如果 **def_data.dspu_services** 未 h 置为 AP_DLUR，则此字段 + ；忽 T 。

def_data.conventional_lu_compression

指定 k 此 PU 相关 D # 规 LU 会话 G 否 k s x 行 }] 压 u 。

AP_NO

> X Z c ；& 对在 # 规 LU 会话 O w 动 D、9 C 此 PU D }] x 行压 u 或 b 压 u 。

AP_YES

如果主机 k s 对 }] x 行压 u，则 }] 压 u & 对 k 此 PU 关 * D # 规

LU 会话启C。如果此值已h置，而Z c ；支V压u（在START_NODE 动词O定义），则 INTERNAL_PU + I功定义，+；支V压u。

def_data.conventional_lu_cryptography

指定k此 PU 相关D#规 LU 会话G否要s会话c加\。

AP_NONE

>XZ c ；&对在#规 LU 会话Ow动D、9C此 PU D｝]x行压u或b压u。

AP_MANDATORY

如果 LU P一<入\钥，则?制性会话c加\ I APPN 执行。否则，|X须I 9C LU D&CL序执行（如果bG PU 集中，则I下N LU 执行）。

AP_OPTIONAL

此值允许加\D9C I主机&CL序以?会话为基础来}动。如果主机k s对一v k此 PU 关*D会话x行加\，则L序D行为XwkAP_MANDATORY 相同。如果主机未k s加\，则其行为XwkAP_NONE 相同。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc

AP_OK

如果因N}错误而9动词；能执行，则L序返回下PN}:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PU_NAME

AP_INVALID_PU_ID
AP_INVALID_DLUS_NAME
AP_INVALID_BKUP_DLUS_NAME
AP_INVALID_CLU_CRYPTOGRAPHY

如果因状，错误而9动词；能执行，则L序返回下PN}:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_PU_ALREADY_DEFINED

AP_CANT_MODIFY_VISIBILITY

如果因Z c P未启动而9动词；能执行，则L序返回下PN}:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_LOCAL_LU

DEFINE_LOCAL_LU 动词k s对> X LU（带指定X性）D定义，或_，如果 LU 已存在，则k s对 LU D **attach_routing_data** X性x行修D。注意，如果C DEFINE_LOCAL_LU 来修D一v现P D定义，则} **attach_routing_data** 字段外D任何N } + ；忽T。

VCB a构

格式 1

```
typedef struct define_local_lu
{
        unsigned short  opcode;           /* verb operation code        */
        unsigned char   reserv2;          /* reserved                   */
        unsigned char   format;           /* format                     */
        unsigned short  primary_rc;       /* primary return code        */
        unsigned long   secondary_rc;     /* secondary return code      */
        unsigned char   lu_name[8];       /* local LU name              */
        LOCAL_LU_DEF_DATA
                        def_data;         /* defined data               */
} DEFINE_LOCAL_LU;

typedef struct local_lu_def_data
{
        unsigned char   description;      /* resource description       */
        unsigned char   lu_alias[8];      /* local LU alias             */
        unsigned char   nau_address;      /* NAU address                */
        unsigned char   syncpt_support;   /* is sync-point supported?   */
        unsigned short  lu_session_limit; /* LU session limit           */
        unsigned char   default_pool;     /* member of default_lu_pool  */
        unsigned char   reserv2;          /* reserved                   */
        unsigned char   pu_name[8];       /* PU name                    */
        unsigned char   lu_attributes;    /* LU attributes              */
        unsigned char   sscp_id[6];       /* SSCP ID                    */
        unsigned char   disable;          /* disable or enable LOCAL LU */
        unsigned char   attach_routing_data;
                                          /* routing data for           */
                                          /* incoming attaches          */
        unsigned char   lu_model;         /* LU model for SDDLU         */
        unsigned char   model_name[7];    /* LU model name              */
                                          /* for SDDLU                  */
        unsigned char   reserv4[16];      /* reserved                   */
} LOCAL_LU_DEF_DATA;
```

VCB a构

格式 0

```
typedef struct define_local_lu
{
        unsigned short  opcode;           /* verb operation code        */
        unsigned char   reserv2;          /* reserved                   */
        unsigned char   format;           /* format                     */
        unsigned short  primary_rc;       /* primary return code        */
        unsigned long   secondary_rc;     /* secondary return code      */
        unsigned char   lu_name[8];       /* local LU name              */
        LOCAL_LU_DEF_DATA
                        def_data;         /* defined data               */
} DEFINE_LOCAL_LU;
```

```
typedef struct local_lu_def_data
{
        unsigned char   description;      /* resource description       */
        unsigned char   lu_alias[8];      /* local LU alias             */
        unsigned char   nau_address;      /* NAU address                */
        unsigned char   syncpt_support;   /* is sync-point supported?   */
        unsigned short  lu_session_limit; /* LU session limit           */
        unsigned char   default_pool;     /* member of default_lu_pool  */
        unsigned char   reserv2;          /* reserved                   */
        unsigned char   pu_name[8];       /* PU name                    */
        unsigned char   lu_attributes;    /* LU attributes              */
        unsigned char   sscp_id[6];       /* SSCP ID                    */
        unsigned char   disable;          /* disable or enable LOCAL LU */
        unsigned char   attach_routing_data;
                                          /* routing data for           */
                                          /* incoming attaches          */
    } LOCAL_LU_DEF_DATA;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_DEFINE_LOCAL_LU

format

j 6 VCB D q = 。 + 此字段 h 置为 c 或一，以指定在O f P v VCB D q = 0 或 1。

lu_name

；定义 > X LU D { F。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。

def_data.description

资源 5 w（返回 = QUERY_LOCAL_LU O）。 | G 一 v 在 > X I 显 > 字符集中 D 一 v 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

def_data.lu_alias

要定义 > X LU D p { 。 | G 一 v 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。 y P 8 v 字 Z 都 G P 效 D， " R X 须 h 置。

def_data.nau_address

LU D网 g I 寻址 % 元 X 址， X 须在 0--255 范围内。非 c 值 5 指 LU G 一从 t LU。 c 值 5 指 LU G 一独 " LU。

def_data.syncpt_support

此字段 & < 终 h 置为 AP_NO， } 非此 LU P 一同 = c 管理器。

def_data.lu_session_limit

LU 支 V D 最大会话 } 。 c 值 m > 无限制。如果 LU G 独 " D，则此字段 I h 置为任何值。如果 LU G 从 t D，则此字段 X 须 h 置为 1。

def_data.default_pool

如果 LU G 从 t LU6.2 缺 ! X D 一 v I 员，则 h 置 AP_YES。

def_data.pu_name

此 LU + 9 C D PU D { F。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。此字段 v I 从 t LU 9 C， R 对独 " LU & h 置为全二 x 制 c 。

def_data.lu_attributes

指定P关 LU D x一=信息。此字段I为值 AP_NONE，或下P值 OR（或Y作）组合中D一v或多v。

AP_DISABLE_PWSUB

{ C > X LU D Z n f代支V。

def_data.sscp_id

指定允许激活此 LU D SSCP D ID。|G一v 6字Z二x制字段。此字段v I从t LU 9C，R对独" LU、或如果 LU I任何 SSCP 激活&h置为全二x制c。

def_data.disable

指> LOCAL LU &{ C还G启C。I通过重新发v带J 1 N }（AP_YES 或 AP_NO）D DEFINE_LOCAL_LU，动，X启C或{ C LU。1一v{ C D LU；启C 1，L序+发v一v NOTIFY（*机）。1一v启C D LU ；{ C 1，L序+发v一v NOTIFY（脱机）。如果 LU ；{ C 1处Z，S状，，则L序+在NOTIFY（脱机）之后发v一v UNBIND。

def_data.attach_routing_data

，S 7 6 }] D类型。

AP_REGISTERED_OR_DEFAULT_ATTACH_MGR

指定+ DYNAMIC_LOAD_INDICATION（I在此>X LU 处=达B务L序（TP）D，S z z）发M=，S管理器（已注a S U此 LU D DLI），或如果；P此 LU D已注a，S管理器，则发M=缺！，S管理器。

AP_REGISTERED_ATTACH_MGR_ONLY

指定v + DYNAMIC_LOAD_INDICATION（I在此>X LU 处=达B务L序（TP）D，S z z）发M=，S管理器（已注a S U此 LU D DLI）。或如果；P此 LU D已注a，S管理器，则\ x，S。

def_data.lu_model

LU D模型类型和号k。此字段v I从t LU 9C，R对独" LU &h置为AP_UNKNOWN。对Z从t LU，|h置为下P值之一:

AP_3270_DISPLAY_MODEL_2
AP_3270_DISPLAY_MODEL_3
AP_3270_DISPLAY_MODEL_4
AP_3270_DISPLAY_MODEL_5
AP_RJE_WKSTN
AP_PRINTER
AP_SCS_PRINTER
AP_UNKNOWN

对Z从t LU，如果 **model_name** 未h置为全二x制c，则此字段+；忽T。如果指定K } AP_UNKNOWN 外D其{值，R主机系统支V SDDLU（自定义从t LU），则Z c + z I一v非k s D PSID NMVT &答，以动，X在主机O定义>X LU。PSID 子向？|含k此字段相对&D机器类型和型号。通过重新发v动词，此字段I动，X x行|D。在 LU 关U和M放0，|D +；起作C。

def_data.model_name

LU D模型{ F。此字段v I从t LU 9C, R对独" LU &h置为二x制c。APPN 检i此字段G否I EBCDIC 字符 A-Z、 0-9 和 @、# 和 $ y组I。

如果此字段未h置为二x制c, R主机系统支V SDDLU, 则Z c + z I一v非k s D PSID NMVT &答, 以动, X在主机O定义> X LU。PSID 子向？| 含此字段中a供D{ F。通过重新发v动词, I对 **def_data.model_name** 动, X x行| D。在 LU 关U和M放0, | D + ；起作C。

返回N数

如果动词I功执行, 则L序返回下P N }:

primary_rc

AP_OK

如果因N }错误而9动词；能执行, 则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_MODEL

AP_INVALID_LU_NAME
AP_INVALID_NAU_ADDRESS
AP_INVALID_SESSION_LIMIT
AP_INVALID_DISABLE

如果因状，错误而9动词；能执行, 则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_PU_NOT_DEFINED

AP_INVALID_LU_NAME
AP_LU_ALREADY_DEFINED
AP_ALLOCATE_NOT_PENDING
AP_LU_ALIAS_ALREADY_USED
AP_PLU_ALIAS_ALREADY_USED
AP_PLU_ALIAS_CANT_BE_CHANGED

如果因Z c P未启动而9动词；能执行, 则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行, 则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

secondary_rc

AP_MEMORY_SHORTAGE

DEFINE_LS

DEFINE_LS C Z定义新D 4 7 >（LS）或修D现P D 4 7 >。此动词a供 LS { F（在{ v Z c中唯一）和此 LS &9 C端Z D{ F。此端Z X须已 9 C DEFINE_PORT 动词定义。X定4 7 }] ，S = 基> a构中。如果4 7 >处Z 4位状，（在发v STOP_LS后）， R自 LS 0一次定义以来，在 DEFINE_LS O指定D **port_name** 未做 | D，则 DEFINE_LS v I C Z修D现P 4 7 > D一v或多v字段。

如需P关 DLC、端Z和4 7 >间关系D | 多信息， k N阅Z 16页D『DLC 过L、端Z和4 7 >』。

LS_DEF_DATA 中大？字段D h置取v Z **adj_cp_type** 字段D值。**adj_cp_type** I 9 C 8 种值（在 **def_data.adj_cp_type** 中做x一= h v），其中 4 种值C Z k相Z 2.1 类（APPN）Z c D 4 7:

- AP_NETWORK_NODE
- AP_END_NODE
- AP_APPN_NODE
- AP_BACK_LEVEL_LEN_NODE

而m 4 种值v C Z带 PU 2.0 类通信？D 4 7:

- AP_HOST_XID3
- AP_HOST_XID0
- AP_DSPU_XID
- AP_DSPU_NOXID。

P 4 种类型D APPN Z c，如下y >

- 一种 APPN 网g Z c，在其 XID3 中 | 含网g { F X制向？（CV），支V " 行 TG，在其 XID3 中h置，网能&位， " 支V 4 7 O D CP-CP 会话。
- 一种 APPN 端Z c，在其 XID3 中 | 含网g { F CV，支V " 行 TG， ；在其 XID3 中h置，网能&位， " 支V 4 7 O D CP-CP 会话。
- 一种外c Z c，在其 XID3 中 | 含网g { F CV，支V " 行 TG， ；在其 XID3 中h置，网能&位， " ；支V CP-CP 会话。
- 一种内c Z c，在其 XID3 中；| 含网g { F CV， ；支V " 行 TG， ；在其 XID3 中h置，网能&位， " ；支V CP-CP 会话。

下P字段对y P 4 7 X须h置:

port_name

adj_cp_type

dest_address

auto_act_supp

disable_remote_act

limited_resource

link_deact_timer

ls_attributes

adj_node_id

local_node_id

target_pacing_count

max_send_btu_size

link_spec_data_len

ls_role

其 | 字段X须 h 置如下:

- 如果 **adj_cp_type** h 置为 AP_NETWORK_NODE、 AP_END_NODE 或 AP_APPN_NODE, 则下 P 字段X须 h 置:

  adj_cp_name

  tg_number

  solicit_sscp_sessions

  dspu_services

  hpr_supported

  hpr_link_lvl_error

  default_nn_server

  cp_cp_sess_support

  use_default_tg_chars

  tg_chars

- 如果 **adj_cp_type** h 置为 AP_BACK_LEVEL_LEN_NODE, 则下 P 字段X须 h 置:

  adj_cp_name

  solicit_sscp_sessions

  dspu_services

  use_default_tg_chars

  tg_chars

- 如果一 v > X PU 要 9 C 4 7（**adj_cp_type** h 置为 AP_HOST_XID3 或 AP_HOST_XID0, 或在 = APPN Z c 4 7 O D **solicit_sscp_sessions** h 置为 AP_YES）, 则下 P 字段X须 h 置:

  pu_name

- 如果一 v 下N PU 要 9 C 4 7, " I PU 集中 a 供服务（**dspu_services** h 置为 AP_PU_CONCENTRATION）, 则下 P 字段X须 h 置:

  dspu_name

- 如果一 v 下N PU 要 9 C 4 7, " I DLUR a 供服务（**dspu_services** h 置为 AP_DLUR）, 则下 P 字段X须 h 置:

  dspu_name

  dlus_name

  bkup_dlus_name

VCB a 构

```
typedef struct define_ls
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   attributes;         /* verb attributes               */
```

```
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* current format is zero        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned char   ls_name[8];         /* name of link station          */
        LS_DEF_DATA     def_data;           /* LS defined data               */
} DEFINE_LS;

typedef struct ls_def_data
{
        unsigned char   description[RD_LEN];
                                            /* resource description          */
        unsigned char   port_name[8];       /* name of associated port       */
        unsigned char   adj_cp_name[17];    /* adjacent CP name              */
        unsigned char   adj_cp_type;        /* adjacent node type            */
        LINK_ADDRESS    dest_address;       /* destination address           */
        unsigned char   auto_act_supp;      /* auto-activate supported       */
        unsigned char   tg_number;          /* Pre-assigned TG number        */
        unsigned char   limited_resource;   /* limited resource              */
        unsigned char   solicit_sscp_sessions;
                                            /* solicit SSCP sessions         */
        unsigned char   pu_name[8];         /* Local PU name (reserved if    */
                                            /* solicit_sscp_sessions is set  */
                                            /* to AP_NO)                     */
        unsigned char   disable_remote_act; /* disable remote activation flag */
        unsigned char   dspu_services;      /* Services provided for         */
                                            /* downstream PU                 */
        unsigned char   dspu_name[8];       /* Downstream PU name (reserved  */
                                            /* if dspu_services is set to    */
                                            /* AP_NONE or AP_DLUR)           */
        unsigned char   dlus_name[17];      /* DLUS name if dspu_services    */
                                            /* set to AP_DLUR                */
        unsigned char   bkup_dlus_name[17]; /* Backup DLUS name if           */
                                            /* dspu_services set to AP_DLUR  */
        unsigned char   hpr_supported;      /* does the link support HPR?    */
        unsigned char   hpr_link_lvl_error; /* does link use link-level      */
                                            /* error recovery for HPR frms?  */
        unsigned short  link_deact_timer;   /* HPR link deactivation timer   */
        unsigned char   reserv1;            /* reserved                      */
        unsigned char   default_nn_server;  /* Use as deflt LS to NN server  */
        unsigned char   ls_attributes[4];   /* LS attributes                 */
        unsigned char   adj_node_id[4];     /* adjacent node ID              */
        unsigned char   local_node_id[4];   /* local node ID                 */
        unsigned char   cp_cp_sess_support; /* CP-CP session support         */
        unsigned char   use_default_tg_chars;
                                            /* Use the default tg_chars      */
        TG_DEFINED_CHARS tg_chars;          /* TG characteristics            */
        unsigned short  target_pacing_count;/* target pacing count           */
        unsigned short  max_send_btu_size;  /* max send BTU size             */
        unsigned char   ls_role;            /* link station role to use      */
                                            /*  on this link                 */
        unsigned char   max_ifrm_rcvd;      /* max number of I-frames rcvd   */
        unsigned short  dlus_retry_timeout; /* DLUS retry timeout            */
        unsigned short  dlus_retry_limit;   /* DLUS retry limit              */
        unsigned char   conventional_lu_compression;
                                            /* Data compression requested for */
                                            /* conventional LU sessions      */
        unsigned char   conventional_lu_cryptography;
                                            /* Cryptography required for     */
                                            /* conventional LU sessions      */
        unsigned char   reserv3;            /* reserved                      */
        unsigned char   retry_flags;        /* conditions LU sessions        */
        unsigned short  max_activation_attempts;
                                            /* how many automatic retries:   */
        unsigned short  activation_delay_timer;
                                            /* delay between automatic retries*/
        unsigned char   branch_link_type;   /* branch link type              */
```

```
        unsigned char   adj_brn_cp_support; /* adjacent BrNN CP support       */
        unsigned char   reserv4[20];        /* reserved                       */
        unsigned short  link_spec_data_len; /* length of link specific data   */
} LS_DEF_DATA;

typedef struct tg_defined_chars
{
        unsigned char   effect_cap;         /* effective capacity             */
        unsigned char   reserve1[5];        /* reserved                       */
        unsigned char   connect_cost;       /* connection cost                */
        unsigned char   byte_cost;          /* byte cost                      */
        unsigned char   reserve2;           /* reserved                       */
        unsigned char   security;           /* security                       */
        unsigned char   prop_delay;         /* propagation delay              */
        unsigned char   modem_class;        /* modem class                    */
        unsigned char   user_def_parm_1;    /* user-defined parameter 1       */
        unsigned char   user_def_parm_2;    /* user-defined parameter 2       */
        unsigned char   user_def_parm_3;    /* user-defined parameter 3       */
} TG_DEFINED_CHARS;

typedef struct link_address
{
        unsigned short  length;             /* length                         */
        unsigned short reserve1;            /* reserved                       */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                            /* address                        */
} LINK_ADDRESS;

typedef struct link_spec_data
{
       unsigned char  link_data[SIZEOF_LINK_SPEC_DATA];

}  LINK_SPEC_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_LS

attributes

动词D t性。此字段G一v位字段。 Z一位|含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

ls_name

4 7 > D { F。|G一v在> X I显>字符集中D一v 8字Z字符串。y P 8 v字Z都G P效D， " R X须h置。

+字段 **ls_name** h置为X b值 “$ANYNET$” （一v ASCII 字符串）_ P下P效果: |通知Z c Y作员h) b M G独 " LU 会话通信？（I AnyNet DLC 7 I传M）&发M = D 4 7 >。如果要s AnyNet 7 I选择，则在 AnyNet DLC O D某一端Z O X须定义一v此{ F D 4 7 >。

def_data.description

资源 5 w（返回 = QUERY_LS、QUERY_PU O）。 | G 一 v 在 > X I 显 > 字符集中 D 一 v 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

def_data.port_name

k 此 4 7 > 关 * D 端 Z D { F。 | G 一 v 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。 y P 8 v 字 Z 都 G P 效 D， " R X 须 h 置。此 _ { 端 Z X 须已 I 一 v DEFINE_PORT 动词定义。

def_data.adj_cp_name

全限定 17 字 Z 相 Z X 制 c { F， " 以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（？ v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人 U q。） 此字段 v J C Z = APPN Z c D 4 7， } 此之外 + ；忽 T。对 Z = APPN Z c D 4 7，此字段 I h 置为全 c， } 非字段 **tg_number** h 置为 1 = 20 范围内 D 某一 } 字，或字段 **adj_cp_type** h 置为 AP_BACK_LEVEL_LEN_NODE。如果此字段 h 置为全 c，则；k 在 XID ；换期间从相 Z Z c S U = D { F x 行对 U 检 i 。如果此字段未 h 置为全 c，则 + k 在 XID ；换期间从相 Z Z c S U = D { F x 行对 U 检 i， } 非 **adj_cp_type** h 置为 AP_BACK_LEVEL_LEN_NODE （在此 i v 下，此字段 C Z j 6 相 Z Z c）。

def_data.adj_cp_type

相 Z Z c 类型。

AP_NETWORK_NODE

指定 Z c 为 APPN 网 g Z c 。

AP_END_NODE

指定 Z c 为 APPN 端 Z c 或外 c Z c 。

AP_APPN_NODE

指定 Z c 为 APPN 网 g Z c 、APPN 端 Z c 或外 c Z c 。 Z c 类型 + 在 XID ；换期间获 C。

AP_BACK_LEVEL_LEN_NODE

指定 Z c 为 back_level_len Z c 。即， | ；发 M XID 中 D X 制 c { F。对 Z 一 v 支 V 独 " LU 会话、 9 C AnyNet DLC D 4 7， X 须指定 AP_BACK_LEVEL_LEN_NODE。

AP_HOST_XID3

指定 Z c 为一 v 主机， " R v 人通信或通信服务器对来自 Z c （ q = 3 XID）D V 询 XID 做 v 响 &。

AP_HOST_XID0

指定 Z c 为一 v 主机， " R v 人通信或通信服务器对来自 Z c （ q = 0 XID）D V 询 XID 做 v 响 &。对 Z 一 v 支 V 从 t LU 会话、 9 C AnyNet DLC D 4 7， X 须指定 AP_HOST_XID0。

AP_DSPU_XID

指定 Z c 为下 N PU， " R v 人通信或通信服务器 + XID ；换 | 括在 4 7 激活中。

AP_DSPU_NOXID

指定 Z c 为下 N PU， " R v 人通信或通信服务器； + XID ；换 | 括在 4 7 激活中。

": 一v至 VRN D 4 7 > <终G动，D，因此未定义。

def_data.dest_address.length

相Z Z c O目D 4 7 > X址D $度。

如果 **def_data.dest_address.length** h置为c， " R此 LS k SATF 类D某一端Z关*，则L序+认为此4 7 > G一v通配4 7 >。b 9 C L序+ LS k任何达=，S匹配，而b些，S k m一v定义D 4 7 >；匹配。

def_data.dest_address.address

相Z Z c O 4 7 > D目D X址。对Z 9 C AnyNet DLC D 4 7， **dest_address** 指定相Z Z c ID 或相Z X制c { F。如果指定相Z Z c ID，则$度X须为4，R X址X须|含 4 字Z . y x制Z c ID（1 字Z i ID，3 字Z PU ID）。如果指定相Z X制c { F，则$度X须为 17，R X址X须|含 EBCDIC 形=D X制c { F，"以 EBCDIC U q R n d。

def_data.auto_act_supp

指定1某一会话要s 1， 4 7 G否I；自动激活。（AP_YES 或 AP_NO）。如果4 7；G=达一v APPN Z c D，则此字段I <终h置为 AP_YES，"对其|N}；P要s。如果4 7 =达某一 APPN Z c，则如果4 7也支V CP-CP 会话，那4此字段；I h置为 AP_YES; v在以下i v下I h置为 AP_YES: 如果一v预先指定D TG }字也已为4 7定义（**tg_number**，" h置为 1 = 20之间D值）。如果 **adj_cp_type** h置为 AP_BACK_LEVEL_LEN_NODE，则b些要s + <终z足，b G因为 **cp_cp_sess_support** 和 **tg_number** 在此i v下+；忽T。

def_data.tg_number

预先指定D TG }字。此字段v J C Z =相Z APPN Z c D 4 7，}此之外+；忽T。如果 **adj_cp_type** h置为 AP_BACK_LEVEL_LEN_NODE，则此字段也+；忽T，"假h h置为 1。对Z =达相Z APPN Z c D 4 7，此字段X须h置为 1 = 20 范围内D值。1 4 7激活1此}字C Z m > 4 7。在此4 7激活期间，v人通信或通信服务器+；S \来自相Z Z c D任何其|}字。为K \ b因预先指定 TG }字D；匹配而<致4 7激活' \，在相Z 4 7 > O D相Z Z c（如果9 C预先指定D TG }字）X须定义相同D TG }字。如果定义K一v预先指定D TG }字，则 **adj_cp_name** 也X须定义（"；能h置为全c），R **adj_cp_type** X须h置为 AP_NETWORK_NODE 或 AP_END_NODE。如果d入c，则 TG }字+；G预先指定D，"+在4 7激活1 x行协L。

def_data.limited_resource

指定1；P会话9 C 4 7 1，G否M放4 7 >。|h置为下P值之一:

AP_NO

4 7；G一v \限资源，"；能自动M放。

AP_YES or AP_NO_SESSIONS

4 7 G一v \限资源，" 1；P活动会话9 C | 1 +自动M放。\限资源4 7 > I配置为 CP-CP 会话支V。（I通过+此字段h置为 AP_YES，+ **cp_cp_sess_support** h置为 AP_YES 5现。）在b种i v下，如果 CP-CP 会话在4 7 O v现，则v人通信或通信服务器+；会Q 4 7 1作一v \限资源对待（R；关U 4 7）。

AP_INACTIVITY

4 7 G一v \限资源，1；P活动会话9 C | 1，或1在指定1间内

（ I **link_deact_timer** 字段指定）４７Ｏ；Ｐ｝］w动1，４７＋自动M放。注意，在非；换＝端ＺＯＤ４７＞；能；配置为＼限资源。

注意，在非；换＝端ＺＯＤ４７＞；能；配置为＼限资源。

＼限资源４７＞Ｉ配置为 CP-CP 会话支Ｖ。（Ｉ通过＋此字段ｈ置为 AP_YES，＋ **cp_cp_sess_support** ｈ置为 AP_YES ５现。）在ｂ种ｉｖ下，如果 CP-CP 会话在４７Ｏｖ现，则ｖ人通信或通信服务器＋；会再次Ｑ４７１作一ｖ＼限资源对待（Ｒ；关Ｕ４７）。注意，如果此字段ｈ置为 AP_INACTIVITY，＋；ＪＣ。

def_data.solicit_sscp_sessions

AP_YES ｋｓ相ＺＺｃ启动 SSCP ｋ＞ＸＸ制ｃ及从ｔ LU 之间Ｄ会话。（在ｂ种ｉｖ下，Ｘ须ｈ置 **pu_name**。）AP_NO ｋｓ在此４７；Ｐｋ SSCP Ｄ会话。此字段ｖＪＣＺ＝一ｖ APPN ＺｃＤ４７，｝此之外＋；忽Ｔ。如果相ＺＺｃ；定义为一ｖ主机（**adj_cp_type** ｈ置为 AP_HOST_XID3 或 AP_HOST_XID0），则ｖ人通信或通信服务器＋＜终ｋｓ主机启动 SSCP ｋ＞ＸＸ制ｃ及从ｔ LU 之间Ｄ会话（同样，Ｘ须ｈ置 **pu_name**）。

如果 **dspu_services** ｈ置为 AP_NONE，则在至一相Ｚ APPN ＺｃＤ４７Ｏ，此字段只能ｈ置为 AP_YES。如果此字段ｈ置为 AP_YES，＂Ｒ此 LS ｙ９ＣＤ DLC 定义为 hpr_only，则 DEFINE_LS ＋在Ｎ｝检ｉ１；＼ｘ，（助返回ｋ为 AP_INVALID_SOLICIT_SSCP_SESS。

def_data.pu_name

如果相ＺＺｃ；定义为一ｖ主机，或＿在至一 APPN ＺｃＤ４７Ｏ **solicit_sscp_sessions** ；ｈ置为 AP_YES，则＋９Ｃ此４７Ｄ＞Ｘ PU Ｄ｛Ｆ。ｂＧ一ｖ８字Ｚ字母｝字Ｄ Ａ 型 EBCDIC 字符串（以一ｖ字母＊头），＂以 EBCDIC ＵｑＲｎｄ。如果相ＺＺｃ；Ｐ；定义为一ｖ主机，而；定义为一ｖ其 **solicit_sscp_sessions** ｈ置为 AP_YES Ｄ APPN Ｚｃ，则此字段＋；忽Ｔ。

def_data.disable_remote_act

指定Ｇ否支Ｖ此４７Ｄ远Ｌ激活（AP_YES 或 AP_NO）。

def_data.dspu_services

指定＞ＸＺｃ在４７中ａ供ｘ下Ｎ PU Ｄ服务。｜ｈ置为下Ｐ之一：

AP_PU_CONCENTRATION

＞ＸＺｃ＋向下Ｎ PU ａ供 PU 集中。

AP_DLUR

＞ＸＺｃ＋向下Ｎ PU ａ供 DLUR 服务。此ｈ置ｖ１＞ＸＺｃＧ网ｇＺｃ１Ｐ效。

AP_NONE

＞ＸＺｃ；向下Ｎ PU ａ供任何服务。

如果此字段ｈ置为 AP_PU_CONCENTRATION 或 AP_DLUR，则Ｘ须同１ｈ置 **dspu_name**。

如果相ＺＺｃ；定义为下Ｎ PU（即，**adj_cp_type** ｈ置为 AP_DSPU_XID 或 AP_DSPU_NOXID），则此字段Ｘ须ｈ置为 AP_PU_CONCENTRATION 或 AP_DLUR。如果 **solicit_sscp_sessions** ；ｈ置为 AP_NO，则在至 APPN ＺｃＤ４７

O I + 此字段 h 置为 AP_PU_CONCENTRATION 或 AP_DLUR。如果相 Z Z c ；定义为主机，则此字段 + ；忽 T。

如果此字段未 h 置为 AP_NONE，" R 此 LS y 9 C D DLC 定义为 hpr_only，则 DEFINE_LS + 在 N } 检 i 1 ； \ x，（助返回 k 为 SP_INVALID_DSPU_SERVICES。

def_data.dspu_name

下 N PU D { F。b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d。

如果 **dspu_services** h 置为 AP_PU_CONCENTRATION 或 AP_DLUR，则此字段 X 须 h 置，} 此之外 + ；忽 T。

def_data.dlus_name

1 至下 N Z c D 4 7 激活 1，DLUR 向其 k s SSCP 服务 D DLUS Z c D { F。C 值 & h 置为全 c，或 I = v A 型 EBCDIC 字符串组 I、中间以一 v EBCDIC c ，S、" 以 EBCDIC U q R n d D 17 字 Z 字符串。（？v { F D $ 度最多 I 为 8 v 字 Z，" ；含 6 入 U q。）如果此字段 h 置为全 c，则 1 4 7 激活 1 全 V 缺！ DLUS （如果已 9 C DEFINE_DLUR_DEFAULTS 动词定义）+ ；k s。如果 **dlus_name** h 置为 c，" R 无全 V 缺！ DLUS，则 1 4 7 激活 1 DLUR + ；启动 SSCP * 系。如果 **dspu_services** 未 h 置为 AP_DLUR，则此字段 + ；忽 T。

def_data.bkup_dlus_name

作为下 N PU 8 份 9 C D DLUS Z c D { F。C 值 & h 置为全 c，或 I = v A 型 EBCDIC 字符串组 I、中间以一 v EBCDIC c ，S、" 以 EBCDIC U q R n d D 17 字 Z 字符串。（？v { F D $ 度最多 I 为 8 v 字 Z，" ；含 6 入 U q。）如果此字段 h 置为全 c，则全 V 8 份缺！ DLUS （如果已 9 C DEFINE_DLUR_DEFAULTS 动词定义）+ 作为此 PU D 8 份 9 C。如果 **dspu_services** 未 h 置为 AP_DLUR，则此字段 + ；忽 T。

def_data.hpr_supported

指定在此 4 7 O G 否支 V HPR（AP_YES 或 AP_NO）。此字段 v J C Z = 一 v APPN Z c D 4 7，} 此之外 + ；忽 T。如果 ；G b 种 i v，则 + 此字段 h 置为 A P _ Y E S + < 致动词在 N } 检 i 1 ； \ x，（助返回 k 为 INVALID_NODE_TYPE_FOR_HPR。

def_data.hpr_link_lvl_error

指定 HPR 通信？G 否 9 C 4 S c 错误恢 4 在此 4 7 O 发 M（AP_YES 或 AP_NO）。如果 **hpr_supported** h 置为 AP_NO，则此 N } + ；忽 T。

def_data.link_deact_timer

\ 限资源 M 放计 1 器（k）。

如果 **limited_resource** ；h 置为 AP_INACTIVITY，则如果在 C 计 1 器期间 ；P }] 在 4 7 O 穿过，4 7 + 自动 M 放。

如果指定 c，则 + 9 C 缺！值 30。否则，最小值为 5。（如果 h 置 | 小 D 值，则指定 D 值 + ；忽 T，" 9 C 5。）如果 **limited_resource** h 置为 AP_NO，则此 N } # t 。

def_data.default_nn_server

指定 4 7 G否 I I 一 v 端 Z c 自动激活，以支 V 至网 g Z c 服务器 D CP-CP 会话。（AP_YES 或 AP_NO）。注意， 4 7 X须定义为支 V CP-CP 会话，以 9 此字段 z 效。

def_data.ls_attributes

指定 P 关相 Z Z c D x 一 = 信息。

def_data.ls_attributes[0]

主机类型。

AP_SNA

j 准 SNA 主机。

AP_FNA

FNA（VTAM-F）主机。

AP_HNA

HNA 主机。

def_data.ls_attributes[1]

b G 一 v 位字段。 | I 能 9 C 值 AP_NO 或任何下 P 4 位 OR（或 Y 作）组合。

AP_SUPPRESS_CP_NAME

至内 c LEN Z c 4 7 D网 g { F CV 抑制选项。如果 h 置此位，则在 k 相 Z Z c D XID ；换中；| 含网 g { F CV。（ } 非 **adj_cp_type** h 置为 AP_BACK_LEVEL_LEN_NODE 或 AP_HOST_XID3，否则此位 + ；忽 T。）

AP_REACTIVATE_ON_FAILURE

如果 4 7 G 活动 D + f 即 ' \， v 人通信或通信服务器 + " T 重新激活 4 7。如果重新激活 4 7 " T ' \，则 4 7 + # V 非活动。

AP_USE_PU_NAME_IN_XID_CVS

如果相 Z Z c ；定义为一 v 主机或在至 APPN Z c 4 7 O D **solicit_sscp_sessions** ；h 置为 AP_YES，R AP_SUPPRESS_CP_NAME 位未 h 置，则在 q = 3 XID O发M D网 g { F CV 中 D 全限定 CP { F + I **def_data.pu_name** 中 a 供 D { F（以 CP D 网 g ID 全限定）代 f 。

def_data.adj_node_id

相 Z Z c D Z c ID。| G 一 v 4 字 Z . y x 制字符串。如果 **adj_cp_type** 指 w 相 Z Z c 为一 v T2.1 Z c，则此字段 + ；忽 T，} 非 | 为非 c； " R **adj_cp_type** h 置为 AP_BACK_LEVEL_LEN_NODE，或相 Z Z c ；在其 XID3 中发 M网 g { F CV。如果 **adj_cp_type** h 置为 AP_HOST_XID3 或 AP_HOST_XID0，则此字段 + < 终；忽 T。如果 **adj_cp_type** h 置为 AP_DSPU_XID R 此字段非 c，则 C 此字段来检 i 下 N PU D m份。如果 **adj_cp_type** h 置为 AP_DSPU_NOXID，则此字段要 4 ；忽 T（如果 **dspu_services** 为 AP_PU_CONCENTRATION），要 4 C Z 向 DLUS j 6 下 N PU （如果 **dspu_services** 为 AP_DLUR）。

def_data.local_node_id

4 7 > O XID 中发 M D Z c ID。| G 一 v 4 字 Z . y x 制字符串。如果此字段 h 置为 c，则 **node_id** + 在 XID ；换中 9 C。如果此字段非 c，则 | + 代 f 此 LS O XID ；换 D 值。

def_data.cp_cp_sess_support

指定G否支V CP-CP 会话（AP_YES 或 AP_NO）。此字段v J C Z = 一v APPN Z c D 4 7， }此之外+ ；忽T。如果 **adj_cp_type** h置为 AP_BACK_LEVEL_LEN_NODE，则此字段也+ ；忽T，"假h h置为 AP_NO。

def_data.use_default_tg_chars

指定G否9 C DEFINE_PORT 动词O a供D缺！ TG X性（AP_YES 或 AP_NO）。如果h置为 AP_YES，则 **tg_chars** 字段+ ；忽T。此字段v J C Z = 一v APPN Z c D 4 7， }此之外+ ；忽T。

def_data.tg_chars

TG X性（N阅Z35页D『DEFINE_CN』）。此字段v J C Z = 一v APPN Z c D 4 7， }此之外+ ；忽T。

def_data.target_pacing_count

1 和 32 767 之间D }字值（|括 1 和 32 767），指>此 TG O BIND D期望w = 窗Z大小。 v 1执行固定，S w = 1，此 }字E P效。 v人通信或通信服务器1 0 ； 9 C此值。

def_data.max_send_btu_size

I从4 7 >发MD最大 BTU 大小。此值C Z协L I在一对4 7 >之间发MD最大 BTU 大小。如果4 7 ； _ P HPR-能&，则此字段X须h置为一v大Z H Z 99 D值。如果4 7 _ P HPR-能&，则此字段X须h置为一v大Z H Z 768 D值。

def_data.ls_role

此4 7 > &；假定为D 4 7 > G +。 | I为 AP_LS_NEG、AP_LS_PRI 或 AP_LS_SEC 中D任一v，以选择I协L、主或(助。此字段也I h置为 AP_USE_PORT_DEFAULTS，以选择在 DEFINE_PORT 动词O配置D值。如果 **dlc_type** 为 AP_TWINAX，则v支V AP_LS_SEC。如果 **dlc_type** 为 AP_ANYNET（R **ls_name** 为 “$ANYNET$”），则；支V AP_LS_PRI。

def_data.max_ifrm_rcvd

XID 发M方在确认0 I S U = D最大 I-帧 }目。

范围: 0 -- 127

如果指定c，则+ 9 C DEFINE_PORT D **max_ifrm_rcvd** D值作为缺！值。

def_data.dlus_retry_timeout

在Z二次和后继"T k在 **def_data.dlus_name** 和 **def_data.bkup_dlus_name** 字段中指定D DLUS 取C * 系间D间t （k）。u < "T和Z一次重T之间D间t <终为1 k。如果指定c，则+ 9 C通过 DEFINE_DLUR_DEFAULTS 配置D缺！值。如果 **def_data.dspu_services** 未h置为 AP_DLUR，则此字段+ ；忽T。

def_data.dlus_retry_limit

在Z一次' \后继续k在 **def_data.dlus_name** 和 **def_data.bkup_dlus_name** 字段中指定D DLUS 取C * 系D最大重T次 }。如果指定c，则+ 9 C通过 DEFINE_DLUR_DEFAULTS 配置D缺！值。如果指定 X’FFFF’，APPN +；确定X x行重T。如果 **def_data.dspu_services** 未h置为 AP_DLUR，则此字段+ ；忽T。

def_data.conventional_lu_compression

指定 k 此 PU 相关D # 规 LU 会话G否 k s x 行 }] 压 u。注意，此字段 v 对其O P 载 LU 0 = 3 通信？D 4 7 P 效。

AP_NO

> X Z c ；&对在 # 规 LU 会话O w 动D、9 C 此 PU D }] x 行压 u 或 b 压 u。

AP_YES

如果主机 k s 对 }] x 行压 u，则 }] 压 u &对 k 此 PU 关 * D # 规 LU 会话启 C。如果此值已 h 置，而 Z c ；支 V 压 u（在 START_NODE 动词O定义），则 4 7 > + I 功定义，+ ；支V压 u。

def_data.conventional_lu_cryptography

指定 # 规 LU 会话G否要 s 会话 c 加 \。此字段 v J C Z 其O P 载 LU 0 = 3 通信？D 4 7。

AP_NONE

会话 c 加 \ ；I L 序执行。

AP_MANDATORY

如果 LU P 一 < 入 \ 钥，则？制性会话 c 加 \ I L 序执行。否则，| X 须 I 9 C LU D & C L 序执行（如果 b G PU 集中，则 I 下N LU 执行）。

AP_OPTIONAL

此值允许加 \ D 9 C I 主机& C L 序以？会话为基础来 } 动。如果主机 k s 对一 v k 此 LS O D 会话 x 行加 \，则 L 序D行为X w k AP_MANDATORY 相同。如果主机未 k s 加 \，则其行为X w k AP_NONE 相同。

def_data.retry_flags

此字段指定 4 7 > I 自动重 T 激活D u 件。| G 一 v 位字段，" I 9 C 任何下 P 4 位 OR（或 Y 作）组合。

AP_RETRY_ON_START

在 " T 激活 1，如果未从远 L Z c S U = 响&，则重 T 4 7 激活。如果在 " T 激活 1，基 > 端 Z G 非活动D，则 L 序 + " T 激活 |。

AP_RETRY_ON_FAILURE

如果在活动和暂挂活动 1 4 S ' \，则重 T 4 7 激活。如果在 " T 激活 1，基 > 端 Z 发 z 故O，则 L 序 + " T 激活 |。

AP_RETRY_ON_DISCONNECT

如果 4 7 I 远 L Z c 以 } # 方 = 停止，则重 T 4 7 激活。

AP_DELAY_APPLICATION_RETRIES

I & C L 序启动D 4 7 激活重 T（9 C START_LS 或 k s = 4 7 激活）+ 9 C **activation_delay_timer** x 行 w =。

AP_INHERIT_RETRY

} K 此字段中D j 志指定 D 重 T u 件外，那些在基 > 端 Z 定义 D **retry_flags** 字段中指定D重 T u 件也 + 9 C。

def_data.max_activation_attempts

此字段；z z 效果，} 非在 **retry_flags** 中至 Y h 置一 v j 志。

此字段指定 1 远 L Z c 无响&或基 > 端 Z ；活动 1， L 序允许重 T D 次 } 。 | | 括自动重 T 和 I & C L 序 } 动 D 激活 " T = _ 。

如果达 = 此极限，则；再 x 行自动重 T。C u 件 I STOP_LS、STOP_PORT、STOP_DLC 或一 v I 功激活 4 位。START_LS 或 OPEN_LU_SSCP_SEC_RQ z z 一 v % v 激活 " T， | 在激活 ' \ 1 ；再重 T。

c m > '无限制'。值 AP_USE_DEFAULTS < 致在 DEFINE_PORT O a 供 D **max_activiation_attempts** D 9 C。

def_data.activation_delay_timer

此字段； z z 效果， } 非在 **retry_flags** 中至 Y h 置一 v j 志。

此字段指定在自动重 T 间 L 序 H 待 D k }，以及如果在 **def_data.retry_flags** 中 h 置 AP_DELAY_APPLICATION_RETRIES 位， I & C L 序 } 动 D 激活 " T 间 D k }。

值 AP_USE_DEFAULTS < 致在 DEFINE_PORT O a 供 D **activiation_delay_timer** D 9 C。

如果指定 c，则 L 序 + 9 C 缺！计 1 器 1 间 30 k。

def_data.branch_link_type

v BrNN。| 指定 4 7 G O 行 4 7 或下行 4 7。v 1 **def_data.adj_cp_type** h 置为 AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE 或 AP_BACK_LEVEL_LEN_NODE 1 此字段 J C。

AP_UPLINK

4 7 为 O 行 4 7。

AP_DOWNLINK

4 7 为下行 4 7。

如果字段 **adj_cp_type** h 置为 AP_NETWORK_NODE，则此字段 X 须 h 置为 AP_UPLINK。

其 | Z c 类型：此字段 + 忽 T。

def_data.adj_brnn_cp_support

v BrNN。| 指定相 Z CP G 否允许为、要 s 为或 { 止为一 v NN（BrNN）；例如，一 v BrNN 显 > 一 v NN D 样子。v 1 字段 **adj_cp_type** h 置为 AP_NETWORK_NODE 或 AP_APPN_NODE（R 在 XID ；换期间获知 D Z c 类型为网 g Z c ） 1，此字段 J C。

AP_BRNN_ALLOWED

相 Z CP 允许（ + ；G X 须）为一 v NN（BrNN）。

AP_BRNN_REQUIRED

相 Z CP 要 s 为一 v NN（BrNN）。

AP_BRNN_PROHIBITED

相 Z CP ；允许为一 v NN（BrNN）。

如果字段 **adj_cp_type** h 置为 AP_NETWORK_NODE，字段 **auto_act_supp** h 置为 AP_YES，则此字段 X 须 h 置为 AP_BRNN_REQUIRED 或 AP_BRNN_PROHIBITED。

其 | Z c 类型：此字段 + 忽 T。

def_data.link_spec_data_len
此字段& < 终 h 置为 c 。

返回N数

如果动词 I 功执行，则L 序返回下 P N }:

primary_rc
AP_OK

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_DEF_LINK_INVALID_SECURITY

AP_INVALID_CP_NAME
AP_INVALID_LIMITED_RESOURCE
AP_INVALID_LINK_NAME
AP_INVALID_LS_ROLE
AP_INVALID_NODE_TYPE
AP_INVALID_PORT_NAME
AP_INVALID_AUTO_ACT_SUPP
AP_INVALID_PU_NAME
AP_INVALID_SOLICIT_SSCP_SESS
AP_INVALID_DLUS_NAME
AP_INVALID_BKUP_DLUS_NAME
AP_INVALID_NODE_TYPE_FOR_HPR
AP_INVALID_TARGET_PACING_COUNT
AP_INVALID_BTU_SIZE
AP_HPR_NOT_SUPPORTED
AP_INVALID_TG_NUMBER
AP_MISSING_CP_NAME
AP_MISSING_CP_TYPE
AP_MISSING_TG_NUMBER
AP_PARALLEL_TGS_NOT_SUPPORTED
AP_INVALID_DLUS_RETRY_TIMEOUT
AP_INVALID_DLUS_RETRY_LIMIT
AP_INVALID_CLU_CRYPTOGRAPHY
AP_INVALID_RETRY_FLAGS
AP_BRNN_SUPPORT_MISSING
AP_INVALID_BRANCH_LINK_TYPE
AP_INVALID_BRNN_SUPPORT

如果因状，错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc

AP_LOCAL_CP_NAME

AP_DEPENDENT_LU_SUPPORTED
AP_DUPLICATE_DEST_ADDR
AP_INVALID_NUM_LS_SPECIFIED
AP_LS_ACTIVE
AP_PU_ALREADY_DEFINED
AP_DSPU_SERVICES_NOT_SUPPORTED
AP_DUPLICATE_TG_NUMBER
AP_TG_NUMBER_IN_USE
AP_CANT_MODIFY_VISIBILITY
AP_INVALID_UPLINK
AP_INVALID_DPWNLINK

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_LU_0_TO_3

此动词定义类型 0、1、2 或 3 D LU。 | 允许 + LU m加 = LU X中。如果X；存在，则m加一 v X。此动词 I C Z 修D现 P 定义D **lu_model**、 **model_name**、**priority**、 **description** 和 **appc_spec_def_data**， + ；能修D其 | 字段。

v 人通信或通信服务器支V ACTLU D隐 = LU 类型 0、1、2 或 3 定义。隐 = 定义；能；> }， + 1 LU I 为非活动 1 I ；} 去。要获C P 关隐 = 定义D信息， I 9 C QUERY_LU_0_TO_3 或注 a LU_0_TO_3_INDICATION。一 v 隐 = LU 定义 I 9 C DEFINE_LU_0_TO_3 x 行重定义， u 件G **lu_name**、**pu_name** 和 **nau_address** G } 确D， R **pool_name** 为全 c（ b 1 LU ；1 作 I Y 作员在 Z 一处配置D那样对待）。

VCB a 构

格式 1

```
typedef struct define_lu_0_to_3
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   lu_name[8];             /* LU name                    */
        LU_0_TO_3_DEF_DATA
                        def_data;               /* defined data               */
} DEFINE_LU_0_TO_3;

typedef struct lu_0_to_3_def_data
{
        unsigned char   description             /* resource description       */
        unsigned char   nau_address;            /* LU NAU address             */
        unsigned char   pool_name[8];           /* LU pool name               */
        unsigned char   pu_name[8];             /* PU name                    */
        unsigned char   priority;               /* LU priority                */
        unsigned char   lu_model;               /* LU model                   */
        unsigned char   sscp_id[6]              /* SSCP ID                    */
        unsigned short  timeout;                /* Timeout                    */
        unsigned char   app_spec_def_data[16]; /* Application Specified Data */
        unsigned char   model_name[7];          /* LU model name for DDDLU    */
        unsigned char   reserv3[17];            /* reserved                   */
} LU_0_TO_3_DEF_DATA;
```

VCB a 构

格式 0

```
typedef struct define_lu_0_to_3
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* attributes                 */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   lu_name[8];             /* LU name                    */
        LU_0_TO_3_DEF_DATA
                        def_data;               /* defined data               */
} DEFINE_LU_0_TO_3;

typedef struct lu_0_to_3_def_data
{
        unsigned char   description             /* resource description       */
```

```
unsigned char   nau_address;           /* LU NAU address             */
unsigned char   pool_name[8];          /* LU pool name               */
unsigned char   pu_name[8];            /* PU name                    */
unsigned char   priority;              /* LU priority                */
unsigned char   lu_model;              /* LU model                   */
unsigned char   sscp_id[6]             /* SSCP ID                    */
unsigned short  timeout;               /* Timeout                    */
unsigned char   app_spec_def_data[16]; /* Application Specified Data */
} LU_0_TO_3_DEF_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_LU_0_TO_3

attributes

动词D t性。此字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+此字段h置为c或一，以指定在O f P v VCB D一v f >。

lu_name

;定义> X LU D{ F。b G一v 8字Z字母}字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC U q R n d。

def_data.description

资源5 w（返回= QUERY_LU_0_TO_3 O）。|G一v在> X I显>字符集中D一v 16字Z字符串。y P 16 v字Z都P效。

def_data.nau_address

LU D网g I寻址%元X址，X须在1--255范围内。

def_data.pool_name

此LU y t LU X D{ F。b G一v 8字Z字母}字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC U q R n d。如果LU ; t Z某一X，则此字段+ ; h置为全二x制c。如果X 1 0 ;存在，则创(一v X。

def_data.pu_name

此LU + 9 C D PU（在DEFINE_LS动词O指定）D{ F。b G一v 8字Z字母}字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC U q R n d。

def_data.priority

在发M =主机1 D LU E先级。|h置为下P值之一:

AP_NETWORK
AP_HIGH
AP_MEDIUM
AP_LOW

def_data.lu_model

LU D模型类型和号k。 | h置为下P值之一:

AP_3270_DISPLAY_MODEL_2
AP_3270_DISPLAY_MODEL_3
AP_3270_DISPLAY_MODEL_4
AP_3270_DISPLAY_MODEL_5
AP_RJE_WKSTN
AP_PRINTER
AP_SCS_PRINTER
AP_UNKNOWN

v J C Z q = 1, 如果 **model_name** 未h置为全二x制c, 则此字段+;忽T。

如果指定K } AP_UNKNOWN 外D其{值, R主机系统支V DDDLU(从t LU D动,定义), 则Z c + z I一v非k s D PSID NMVT &答, 以动, X在主机O定义> X LU。对Z q = 1, PSID 子向? | 含k此字段相对&D机器类型和型号。通过重新发v动词, 此字段I动, X x行 | D。在 LU 关U和M放0, | D +;起作C。

def_data.sscp_id

此字段指定允许激活此 LU D SSCP D ID。 | G一v 6 字Z二x制字段。如果字段h置为二x制c, 则 LU 也许I I任何 SSCP 激活。

def_data.timeout

LU D指定, 1值(k)。如果a供K某一, 1值, " R LU D C户在OPEN_LU_SSCP_SEC_RQ O指定K **allow_timeout** (或_, 如为 PU 集中, 则在下N LU O定义), 则如在C指定1间内 PLU-SLU 会话#V非活动, R存在下P u件之一, 则 LU +;M放:

- 会话在\限资源4 7 O传]
- m一&C要在此会话再次9 C 0 9 C LU

如果, 1 h置为c, 则 LU ;会M放。

def_data.app_spec_def_data

&C L序指定定义}]。此字段; I v人通信或通信服务器b M, +;存储" f即返回= QUERY_LU_0_TO_3 动词O。

def_data.model_name

v人通信或通信服务器检i此字段G否I EBCDIC 字符 A-Z、 0-9 和 @、#和 $ y组I。如果此字段未h置为全二x制c, R主机系统支V DDDLU (从t LU D动,定义), 则Z c + z I一v非k s D PSID NMVT &答, 以动, X在主机O定义> X LU。PSID 子向? + | 含此字段中a供D{ F。通过重新发v动词, 此字段I动, X x行 | D。在 LU 关U和M放0, | D +;起作C。

返回N数

如果动词I功执行, 则L序返回下P N }:

primary_rc

AP_OK

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LU_NAME

AP_INVALID_PU_NAME
AP_INVALID_PU_TYPE
AP_PU_NOT_DEFINED
AP_LU_ALREADY_DEFINED
AP_LU_NAU_ADDR_ALREADY_DEFD
AP_CANT_MODIFY_VISIBILITY

如果因状，错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_INVALID_PU_NAME

AP_INVALID_PU_TYPE
AP_PU_NOT_DEFINED
AP_LU_NAME_POOL_NAME_CLASH
AP_LU_ALREADY_DEFINED
AP_LU_NAU_ADDR_ALREADY_DEFD

如果因系统； P ("（ _ P从 t LU 支V ）而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_INVALID_VERB

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c 停止而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_LU_0_TO_3_RANGE

此动词允许在某一指定 NAU 范围内定义多v LU。Z c Y作员a供一v基{和一v NAU 范围。LU { F I +基{和 NAU X址组合在一起z I。此动词;能C Z修D现P定义。

例如，一v基{ LUNME k NAU 范围 1-4 组合在一起，+定义 LU LUNME001、LUNME002、 LUNME003 和 LUNME004。一v基{D $度小Z 5 v非nd字符+9C LU { F D $度小Z 8 v字符。b 1， v人通信或通信服务器+x行Rnd以达= 8 v字符。

VCB a构

格式 1

```
typedef struct define_lu_0_to_3_range
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   base_name[5];           /* base name                  */
        unsigned char   reserv3;                /* reserved                   */
        unsigned char   description;            /* resource description       */
        unsigned char   min_nau;                /* minimum NAU address        */
        unsigned char   max_nau;                /* maximum NAU address        */
        unsigned char   pool_name[8];           /* LU pool name               */
        unsigned char   pu_name[8];             /* PU name                    */
        unsigned char   priority;               /* LU priority                */
        unsigned char   lu_model;               /* LU model                   */
        unsigned char   sscp_id[6];            /* SSCP ID                   */
        unsigned short  timeout;                /* Timeout                    */
        unsigned char   app_spec_def_data[16]; /* application specified data */
        unsigned char   model_name[7];          /* LU model name for DDDLU    */
        unsigned char   name_attributes;       /* Attributes of base name    */
        unsigned char   base_number;           /* Base number for LU names   */
        unsigned char   reserv3[15];            /* reserved                   */
} DEFINE_LU_0_TO_3_RANGE;
```

VCB a构

格式 0

```
typedef struct define_lu_0_to_3_range
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   base_name[5];           /* base name                  */
        unsigned char   reserv3;                /* reserved                   */
        unsigned char   description;            /* resource description       */
        unsigned char   min_nau;                /* minimum NAU address        */
        unsigned char   max_nau;                /* maximum NAU address        */
        unsigned char   pool_name[8];           /* LU pool name               */
        unsigned char   pu_name[8];             /* PU name                    */
        unsigned char   priority;               /* LU priority                */
        unsigned char   lu_model;               /* LU model                   */
```

```
        unsigned char   sscp_id[6];          /* SSCP ID                     */
        unsigned short  timeout;              /* Timeout                     */
        unsigned char   app_spec_def_data;    /* application specified data */
} DEFINE_LU_0_TO_3_RANGE;
```

提供N数

＆C L序a供下P N }:

opcode

AP_DEFINE_LU_0_TO_3_RANGE

attributes

动词D t性。此字段G一v位字段。 Z一位|含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 +此字段h置为c或一，以指定在O f P v VCB D一v f >。

base_name

基> LU { F。 b G一v 5字Z字母}字D A型EBCDIC字符串（以一v字母*头）， "以EBCDIC U q R n d。此基{之后=加3 v A型EBCDIC }字字符， m >在NAU范围内？v LU NAU X址D . x制值。

此字段在字段 **name_attributes** 中； P h置D位。 h置位+ | D此字段D意义。

description

资源5 w（返回= QUERY_LU_0_TO_3 O）。此字段D $度＆为4 v字Z D 6 }， R；为c。

min_nau

范围内D最小NAU X址。 | I为1至255（ |括1和255）。

max_nau

范围内D最大NAU X址。 | I为1至255（ |括1和255）。

pool_name

此LU y t LU X D { F。 b G一v 8字Z字母}字D A型EBCDIC字符串（以一v字母*头）， "以EBCDIC U q R n d。如果LU ; t Z某一X，则此字段+；h置为全二x制c。

pu_name

此LU 9 C D PU（在DEFINE_LS动词O指定）D { F。 b G一v 8字Z字母}字D A型EBCDIC字符串（以一v字母*头）， "以EBCDIC U q R n d。

priority

在发M =主机1 D LU E先级。 | h置为下P值之一:

AP_NETWORK
AP_HIGH
AP_MEDIUM
AP_LOW

lu_model

LU D模型类型和号 k。 | h 置为下 P 值之一:

AP_3270_DISPLAY_MODEL_2
AP_3270_DISPLAY_MODEL_3
AP_3270_DISPLAY_MODEL_4
AP_3270_DISPLAY_MODEL_5
AP_RJE_WKSTN
AP_PRINTER
AP_SCS_PRINTER
AP_UNKNOWN

v J C Z q = 1, 如果 **model_name** 未 h 置为全二 x 制 c，则此字段 + ；忽 T。

如果指定K } AP_UNKNOWN 外D其 { 值, R主机系统支V DDDLU（从 t LU D动，定义），则Z c + z I 一 v 非 k s D PSID NMVT &答, 以动，X 在主机O定义 > X LU。对Z q = 1, PSID 子向？| 含 k 此字段相对& D机器类型和型号。通过重新发 v 动词, 此字段 I 动，X x 行 | D。在 LU 关U和M放 0， | D + ；起作C。

lu_0_to_3_detail.def_data.sscp_id

此字段指定允许激活此 LU D SSCP D ID。 | G一 v 6 字Z二 x 制字段。如果字段 h 置为二 x 制 c，则 LU 也许 I I 任何 SSCP 激活。

lu_0_to_3_detail.def_data.timeout

LU D指定，1 值（ k ）。如果 a 供K某一，1 值, " R LU D C户在 OPEN_LU_SSCP_SEC_RQ O指定K **allow_timeout** （或 _，如为 PU 集中, 则在下N LU O定义），则如在C指定 1 间内 PLU-SLU 会话 # V非活动, R 存在下P u 件之一，则 LU + ；M放:

- 会话在 \ 限资源 4 7 O传]
- m一& C要在此会话再次 9 C 0 9 C LU

如果，1 h 置为 c，则 LU ；会M放。

model_name

v 人通信或通信服务器检 i 此字段G否 I EBCDIC 字符 A-Z、 0-9 和 @、# 和 $ y 组 I。如果此字段未 h 置为全二 x 制 c，R主机系统支V SDDLU （自定义从 t LU），则Z c + z I 一 v 非 k s D PSID NMVT &答, 以动，X在主机O定义 > X LU。PSID 子向？+ | 含此字段中 a 供D { F。

name_attributes

此位字段修D y a 供 **base_name** D b M和C法。此字段 I 9 C c 值或任何下 P 4 位 OR（或Y作）组合。

AP_USE_HEX_IN_NAME

如果 h 置此位，则 **base_name** D b M修D如下:

b G一v 6 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）, "以 EBCDIC U q R n d。基{之后=加 2 v EBCDIC 字符, m>在 NAU 范围内？v LU NAU X址D . y x制值。

AP_USE_BASE_NUMBER
如果h置此位，则 **base_name** D b M修D如下:

b G一v 5 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）, "以 EBCDIC U q R n d。此基{之后=加 3 v EBCDIC } 字字符, m>在范围内 LU D . x制w引, |以 **base_number** *头"以 (**base_name + max_nau -- min_nau**) a x。

base_number
如果在 **name_attributes** 中未h置 AP_USE_BASE_NUMBER 位，则此字段+；忽T。否则，此字段+4以0 h v D方=修D **base_name** D b M。合法值从 0 = (255 -- **max_nau + min_nau**)。

app_spec_def_data
&C L序指定定义}]。此字段；I v人通信或通信服务器b M, +；存储"f即返回= QUERY_LU_0_TO_3 动词O（对范围内D？v LU 返回相同D}]）。

返回N数

如果动词I功执行，则L序返回下P N}:

primary_rc
AP_OK

如果因N}错误而9动词；能执行，则L序返回下P N}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_BASE_NUMBER

AP_INVALID_LU_MODEL
AP_INVALID_LU_NAME
AP_INVALID_NAME_ATTRIBUTES
AP_INVALID_NAU_ADDRESS
AP_INVALID_PRIORITY

如果因状，错误而9动词；能执行，则L序返回下P N}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PU_NOT_DEFINED

AP_INVALID_PU_NAME
AP_INVALID_PU_TYPE
AP_LU_NAME_POOL_NAME_CLASH
AP_LU_ALREADY_DEFINED

AP_LU_NAU_ADDR_ALREADY_DEFD
AP_IMPLICIT_LU_DEFINED
AP_CANT_MODIFY_VISIBILITY

如果因系统；P（"（_P从t LU 支V）而9动词；能执行，则L序返回下PN}:

primary_rc
AP_INVALID_VERB

如果因Z c P未启动而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_LU_POOL

此动词C Z定义 LU X或向现P X中m加 LU。；m加D LU X须已C DEFINE_LU_0_TO_3 动词或 DEFINE_LU_0_TO_3_RANGE 动词定义。LU 在同一1 L只能t Z一v LU X。如果指定D LU 已t Z某一X，则|G+从现P X中移动=；定义D X中。！管对Z一v X中 LU D总}；P限制，+同一1 L至多只能P 10 v LU m加=一v X中。

VCB a构

```
typedef struct define_lu_pool
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   pool_name[8];           /* LU pool name               */
        unsigned char   description[RD_LEN];    /* resource description       */
        unsigned char   reserv3[4];             /* reserved                   */
        unsigned short  num_lus;                /* number of LUs to add       */
        unsigned char   lu_names[10][8];        /* LU names                   */
} DEFINE_LU_POOL;
```

提供N数

&C L序a供下P N}:

opcode

AP_DEFINE_LU_POOL

attributes

动词D t性。此字段G一v位字段。Z一位|含要定义资源D I见性，"k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

pool_name

b些 LU y t D X D{ F。此{ F G一v 8字Z字符串，以U q R n d。|I为一v EBCDIC 字符串或在>X I显>字符集中D一v字符串。

description

资源5 w（返回= QUERY_LU_POOL O）。此字段D $度&为 4 v字Z D 6}， R；为c。

num_lus

要m加 LU D}目，在 0 = 10 之间。

lu_names

m加=X中D LU D{ F。？v{ F G一v 8字Z字母}字D A 型 EBCDIC字符串（以一v字母*头），"以 EBCDIC U q R n d。

返回N数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LU_NAME

AP_INVALID_NUM_LUS
AP_INVALID_POOL_NAME

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_LU_NAME_POOL_NAME_CLASH

AP_INVALID_POOL_NAME

如果因系统；P ("（ _ P 从 t LU 支 V ）而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_INVALID_VERB

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_MODE

DEFINE_MODE 动词定义一组，网X性，以指定x一v X定D方=（或会话组）。此动词也ICZ修D一v以0定义方=D任何字段。如果重定义 SNASVCMG 方=，其 **mode_name** 和 **cos_name** ；能修D。CPSVCMG 方=；能重定义。

DEFINE_MODE 动词也ICZ定义缺！ COS，未知方=+3 d=此缺！ COS。方法G + **mode_name** h置为全c。缺！ COS 一*<为 #CONNECT。

"**:** ！管方=X须在您D网gZc和（I能X）伙iZc处定义，+您仍X须定义yP要在>X9CD方=。如果发v一v ALLOCATE 指定K一vP未定义D方=，则Zc+9C在 DEFINE_DEFAULTS 动词O指定D模型缺！方=DX性。如果；P指定b样D模型，则+9CUW方=模型DX性。

VCB a构

```
typedef struct define_mode
{
        unsigned short  opcode;                 /* verb operation code          */
        unsigned char   reserv2;                /* reserved                     */
        unsigned char   format;                 /* format                       */
        unsigned short  primary_rc;             /* primary return code          */
        unsigned long   secondary_rc;           /* secondary return code        */
        unsigned char   mode_name[8];           /* mode name                    */
        unsigned short  reserv3;                /* reserved                     */
        MODE_CHARS      mode_chars;             /* mode characteristics         */
} DEFINE_MODE;

typedef struct mode_chars
{
        unsigned char   description[RD_LEN]
                                                /* resource description         */
        unsigned short  max_ru_size_upp;        /* max RU size upper bound      */
        unsigned char   receive_pacing_win;     /* receive pacing window        */
        unsigned char   default_ru_size;        /* default RU size to maximize  */
                                                /* performance                  */
        unsigned short  max_neg_sess_lim;       /* max negotiable session limit */
        unsigned short  plu_mode_session_limit; /* LU-mode session limit        */
        unsigned short  min_conwin_src;         /* min source contention winner */
                                                /* sessions                     */
        unsigned char   cos_name[8];            /* class-of-service name        */
        unsigned char   cryptography;           /* cryptography                 */
        unsigned char   compression;            /* compression                  */
        unsigned short  auto_act;               /* initial auto-activation count*/
        unsigned short  min_conloser_src;       /* min source contention loser  */
        unsigned short  max_ru_size_low         /* maximum RU size lower bound  */
        unsigned short  max_receive_pacing_win;
                                                /* maximum receive pacing window*/
        unsigned char   max_compress_lvl;       /* maximum compression level    */
        unsigned char   max_decompression_lvl;  /* maximum decompression level  */
        unsigned char   comp_in_series;         /* support for LZ and RLE       */
        unsigned char   reserv4[24];            /* reserved                     */
} MODE_CHARS;
```

提供N数

&CL序a供下PN}:

opcode

AP_DEFINE_MODE

format

j 6 VCB D q = 。 + 此字段 h 置为 c 或一，以指定在O f P v VCB D f > 。

mode_name

方 = { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d 。如果此字段 h 置为全 c，则缺！ COS h 置为 **mode_chars.cos_name**， R y P 其 | D **mode_chars** 字段 + ；忽 T 。

mode_chars.compression

指定对激活会话（ 9 C 此方 = ）9 C 压 u 。

AP_COMP_PROHIBITED

在此方 = 会话 O ；支 V RLE 压 u 。

AP_COMP_REQUESTED

在此方 = 会话 O 支 V RLE 压 u，" R k s RLE 压 u（ + ；? 制）。

mode_chars.max_ru_size_upp

在此方 = 会话 O 发 M 和 S U RU 最大大小 D O g 。 1 最大 RU 大小在会话激活期间协 L 1， 9 C 此值。范围为 256--61440。如果 **default_ru_size** h 置为 AP_YES，则此字段 + ；忽 T 。

mode_chars.receive_pacing_win

此方 = 会话 D 会话 w = 窗 Z 。对 Z 固定 w =，此值指定 S U w = 窗 Z 。对 Z 自 J & w =，此值作为 W 选窗 Z 大小 9 C 。注意， v 人通信或通信服务器 + < 终 9 C 自 J & w =， } 非相 Z Z c 指定 | ；支 V 自 J & w = 。范围为 1--63。；允许 c 值。

mode_chars.default_ru_size

指定 G 否 9 C 最大 RU 大小 D 缺！O g 。如果此 N } 指定 AP_YES，则 **max_ru_size_upp** + ；忽 T，" R 最大 RU 大小 D O g ；h 置为 4 7 BTU 大小减去 TH 和 RH D 大小。

AP_YES
AP_NO

mode_chars.max_neg_sess_lim

在任何 > X LU 和伙 i LU 间此方 = y 允许 D 最大会话 } 。如果指定 c 值，则；存在隐 = CNOS ；换。范围为 0--32 767。

mode_chars.plu_mode_session_limit

此方 = D 缺！会话限制。 | 限制在任一 > X LU 和伙 i LU 对间此方 = 会话 D } 目。 1 CNOS （会话；换 } ）；换隐 = 启动 1 9 C 此值。如果指定 c 值，则；存在隐 = CNOS ；换。范围为 0--32 767。

mode_chars.min_conwin_src

I I 任一 9 C 此方 = D > X LU 激活 D y C $ 方会话 D 最小 } 目。 1 CNOS（会话；换 } ）；换隐 = 启动 1 9 C 此值。如果指定 c 值，则；存在隐 = CNOS ；换。范围为 0--32 767。

mode_chars.cos_name

1 激活此方 = D 会话 1 y k s 服务级 D { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d 。

mode_chars.cryptography

指定 G 否 X 须 9 C 会话 c 加 \（AP_NONE 或 AP_MANDATORY）。

mode_chars.compression

指定对激活会话（ 9 C此方 = ） 9 C压 u 。

AP_COMP_PROHIBITED

在此方 = 会话O ；支V压 u 。

AP_COMP_REQUESTED

在此方 = 会话O支V压 u ， " R k s 压 u （ + ； ? 制）。

如果 **format** 字段 h 置为 0, 则压 u 和 b 压 u D级 p h 置为Z c y 支V D最大值。

如果 **format** 字段 h 置为 1, 则压 u 和 b 压 u D最 _ 级 p I **max_compress_lvl** 和 **max_decompress_lvl** 字段定义。

mode_chars.auto_act

指定自动激活D此方 = 会话D } ? 。 1 会话；换 } （CNOS）；换隐 = 启动 1 9 C此值。

范围为 0-32767。

mode_chars.min_consloser_src

指定 I 任一此方 = > X LU 激活D y C ：方会话D最小 } 目。 1 CNOS （会话；换 } ）；换隐 = 启动 1 9 C此值。范围为 0-32767。

mode_chars.max_ru_size_low

指定在此方 = 会话O发M和S U RU 最大大小D下 g 。 1 最大 RU 大小在会话激活期间协L 1 , 9 C此值。范围为 256-61140。

c 值m > ； P 下 g 。

如果 **default_ru_size** h 置为 AP_YES, 则此字段 + ；忽T。

mode_chars.max_receive_pacing_win

指定此方 = 会话D会话最大w = 窗Z。对Z自 J & w = , 此值C Z限制 | Z h D S U w = 窗Z。对Z固定w = , ； 9 C此字段。注意, L 序 + < 终 9 C自 J & w = , } 非相Z Z c 指定 | ；支V自 J & w = 。范围为 0-32767。

c 值m > ； P O g 。

mode_chars.max_compress_lvl

L 序 " T x 行协L D、 Z c y 支V D、对 }] w D最大压 u 级。

AP_NONE
AP_RLE_COMPRESSION
AP_LZ9_COMPRESSION
AP_LZ10_COMPRESSION
AP_LZ12_COMPRESSION

配置D压 u 级；能大Z Z c y 支V D级 p （在 START_NODE O D **max_compress_lvl** 字段中指定）。注意, 如果 9 C非扩 9 BIND 对压 u x 行协L , 则压 u 级 + h 置为 RLE 压 u 。

mode_chars.max_decompress_lvl

L 序 " T x 行协L D、 Z c y 支V D、对 }] w D最大 b 压 u 级。

AP_NONE
AP_RLE_COMPRESSION

AP_LZ9_COMPRESSION
AP_LZ10_COMPRESSION
AP_LZ12_COMPRESSION

配置D压u级；能大Z Z c y支V D级p（在 START_NODE O D **max_compress_lvl** 字段中指定）。注意，如果9 C非扩9 BIND 对压u x行协L，则b压u级+ h置为 LZ9 压u。

mode_chars.comp_in_series

指定G否允许在 RLE 压u后9 C LZ 压u。如果此字段h置为 AP_YES，则 **max_compress_lvl** X须h置为 AP_LZ9_COMPRESSION、AP_LZ10_COMPRESSION 或 AP_LZ12_COMPRESSION。

AP_YES

AP_NO

如果Z c；配置为；支V RLE 和 LZ 压u（在 START_NODE O D **comp_in_series** 字段中指定），则此字段；能h置为 AP_YES。

返回N数

如果动词I功执行，则L序返回下P N }：

primary_rc

AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }：

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_COS_NAME

AP_CPSVCMG_ALREADY_DEFD
AP_INVALID_CNOS_SLIM
AP_INVALID_COS_SNASVCMG_MODE
AP_INVALID_DEFAULT_RU_SIZE
AP_INVALID_MAX_NEGOT_SESS_LIM
AP_INVALID_MAX_RU_SIZE_UPPER
AP_INVALID_MAX_RU_SIZE_LOW
AP_RU_SIZE_LOW_UPPER_MISMATCH
AP_INVALID_COMPRESSION
AP_INVALID_MIN_CONWINNERS
AP_INVALID_MIN_CONLOSERS
AP_INVALID_MIN_CONTENTION_SUM
AP_INVALID_MODE_NAME
AP_INVALID_RECV_PACING_WINDOW
AP_INVALID_MAX_RECV_PACING_WIN
AP_INVALID_DEFAULT_RU_SIZES
AP_INVALID_SNASVCMG_MODE_LIMIT

AP_MODE_SESS_LIM_EXCEEDS_NEG
AP_INVALID_CRYPTOGRAPHY
AP_INVALID_MAX_COMPRESS_LVL
AP_INVALID_MAX_DECOMPRESS_LVL
AP_INVALID_COMP_IN_SERIES

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

重定e的'果

以下G？v字段D重定义效果:

description
|新 **description** 返回=后继D QUERY_MODE 动词O。

compression

max_compress_lvl

max_decompress_lvl

comp_in_series

cryptography

max_ru_size_upp

receive_pacing_win

default_ru_size

max_ru_size_low

max_receive_pacing_win
|新值C Z y P后继D、此方=D会话激活" T， "返回= y P后继QUERY_MODE 动词O。|D；0响任何现P活动会话。

max_neg_sess_lim

plu_mode_session_limit

min_conwin_src

auto_act

min_conloser_src
在下一v CNOS |n（>X启动或远L启动）0，|新值；&C Z某一>X LU 对或伙i LU 对。在下一v CNOS |n 1， I值+返回= QUERY_MODE 动词。

cos_name

| 新值C Z y P后继D、此方= D会话激活" T， "返回= y P后继QUERY_MODE 动词O。| D；0响任何现P活动会话。| 新值还C Z y P k对 COS 3 d Y作D后继方=（例如，如果此Z c G一v网g Z c， R a供对COS 3 d服务或其服务D端Z c D方=），"返回= y P后继QUERY_MODE_TO_COS_MAPPING 动词O。

"：一v隐=方=定义I I DEFINE_MODE 显= X 5现。b I后继 QUERY_MODE 动词返回 **implicit set** 为 AP_NO 来反3。

DEFINE_PARTNER_LU

DEFINE_PARTNER_LU 动词定义在一 v > X LU 和一 v 伙 i LU 间 LU-LU 会话中伙 i LU D N }。或 _， DEFINE_PARTNER_LU I C Z 修 D 已为伙 i LU 定义 D y P N }， **fqplu_name** 和 **plu_alias** } 外。

VCB a 构

```
typedef struct define_partner_lu
{
        unsigned short  opcode;             /* verb operation code          */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   format;             /* format                       */
        unsigned short  primary_rc;         /* primary return code          */
        unsigned long   secondary_rc;       /* secondary return code        */
        PLU_CHARS       plu_chars;          /* partner LU characteristics   */
} DEFINE_PARTNER_LU;

typedef struct plu_chars
{
        unsigned char   fqplu_name[17];     /* fully qualified partner      */
                                            /* LU name                      */
        unsigned char   plu_alias[8];       /* partner LU alias             */
        unsigned char   description[RD_LEN];
                                            /* resource description         */
        unsigned char   plu_un_name[8];     /* partner LU uninterpreted name */
        unsigned char   preference          /* routing preference           */
        unsigned short  max_mc_ll_send_size; /* max MC send LL size         */
        unsigned char   conv_security_ver;  /* already_verified accepted?   */
        unsigned char   parallel_sess_supp; /* parallel sessions supported? */
        unsigned char   reserv2[8];         /* reserved                     */
} PLU_CHARS;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_DEFINE_PARTNER_LU

format

j 6 VCB D q =。+ 此字段 h 置为 c，以指定在 O f P v VCB D f >。

plu_chars.fqplu_name

伙 i LU D 全限定 { F。{ F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。(? v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 入 U q。)

plu_chars.plu_alias

伙 i LU D p {。| G 一 v 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。对 Z 一 v ；P p { k 之关 * D 伙 i LU，此字段 I h 置为全 c。

plu_chars.description

资源 5 w (返回 = QUERY_PARTNER_LU 和 QUERY_PARTNER_LU_DEFINITION O)。| G 一 v 在 > X I 显 > 字符集中 D 一 v 16 字 Z 字符串。y P 16 v 字 Z 都 P 效。

plu_chars.plu_un_name

伙 i LU D 非 b M { F。b G 一 v 8 字 Z A 型 EBCDIC 字符串。

plu_chars.max_mc_ll_send_size

在伙 i LU 处， 3象会话服务发M和S U D最大 LL 记<大小。范围: 1-32 767（32 767 通过+此字段h置为 0 指定）。

plu_chars.preference

C Z至此伙 i LU D会话激活DW选7 I 选择协议。此字段I 9 C下P值:

AP_NATIVE

v 9 C >机（APPN）7 I 选择协议。

AP_NONNATIVE

v 9 C非>机（AnyNet）协议。

AP_NATIVE_THEN_NONNATIVE

先" T >机（APPN）协议，如果无法R =伙 i LU 则9 C非>机（AnyNet）协议重T会话激活。

AP_NONNATIVE_THEN_NATIVE

先" T非>机（AnyNet）协议，如果无法R =伙 i LU 则9 C >机（APPN）协议重T会话激活。

AP_USE_DEFAULT_PREFERENCE

在Z c启动1 9 C定义D缺！W选项。（I I QUERY_NODE 重w C。）

": v 1 Z c Y作员h) P一 AnyNet DLC， R存在一v已定义D AnyNet 4 7 > 1， 非>机7 I 选择E P意义。（N阅Defined_LS）。

plu_chars.conv_security_ver

指定G否Z权伙 i LU 代m > X LU 对 **user_ids** x行验证，即伙 i LU G否I在一v ，S k s中h置已验证指>符（AP_YES 或 AP_NO）。

plu_chars.parallel_sess_supp

指定伙 i LU G否支V "行会话（AP_YES 或 AP_NO）。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_ANYNET_NOT_SUPPORTED

AP_DEF_PLU_INVALID_FQ_NAME
AP_INVALID_UNINT_PLU_NAME

如果因状，错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_PLU_ALIAS_CANT_BE_CHANGED

AP_PLU_ALIAS_ALREADY_USED

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

重定e的'果

以下G？v字段D重定义效果:

fqplu_name

；能|D。

plu_alias

如果一v以0 D DEFINE_PARTNER_LU 以一v；同D **plu_alias** 发v，则DEFINE_PARTNER_LU + ' \。如果一v以0 D DEFINE_PARTNER_LU 以一v全c D **plu_alias** 发v，则重定义+；S \ " + 0响y P现P RLU记<。如果以0未发v过 DEFINE_PARTNER_LU，则指定D **plu_alias** + = 4 = y P相&隐=定义D伙i LU 记<中，}非指定全c，在此i v下隐= **plu_aliases** ；D动。

"： 如果某些&C L序已从一v O早D APPC 动词获C K隐= **plu_alias**，" C Z一v后继D ALLOCATE，则发v带非c **plu_alias** D DEFINE_PARTNER_LU I <致b些运行&C L序' \。

description

|新 **description** 返回=后继D QUERY_PARTNER_LU 动词O。

plu_un_name

|新D **plu_un_name** C Z y P至此伙i LU D后继会话激活k s，"返回= y P后继 QUERY_PARTNER_LU 动词O。

preference

|新D **preference** C Z y P至此伙i LU D后继会话激活k s，"返回= y P后继 QUERY_PARTNER_LU 动词O。

max_mc_ll_send_size

|新D **preference** C Z y P至此伙i LU D后继会话激活k s（即9在现P会话O）。|D；0响现P对话。|新值返回= y P后继D QUERY_PARTNER_LU 动词O。

conv_security_ver

在 > X LU 和伙 i LU 间D会话 } 5 为 c 之 0， | 新值；C Z 某一 > X LU。BIND 和 RSP(BIND) + 9 C I h 置 w 动， R I 值 + 返回 = QUERY_PARTNER_LU k s 中，直 = 会话 } 5 为 c。 b G因为如果后继会话D 2 全性支V k 现 P 活动会话D 2 全性支V；同，伙 i LU I 以 \ x 后继会话激活 " T。

parallel_sess_supp

1 k **conv_security_ver** 一起 9 C 1，在 > X LU 和指定伙 i LU 间D会话 } 5 为 c 之 0， | 新值；C Z 某一 > X LU。 b G为K \ b h 计D LU6.2 会话一致性检 i 问 b。

"： 一 v 隐 = 方 = 定义 I I DEFINE_PARTNER_LU 显 = X 5 现。 b I 后继 QUERY_PARTNER_LU 动词返回 **implicit set** AP_NO 来反 3。

```
        unsigned char   ls_role;              /* initial link station role     */
        unsigned char   retry_flags;          /* conditions for automatic      */
                                              /* retries                       */
        usigned char    max_activation_attempts;
                                              /* how many automatic retries?   */
        unsigned char   adtivation_delay_timer;
                                              /* delay between automatic       */
                                              /* retries                       */
        unsigned char   reserv1[10];          /* reserved                       */
        unsigned char   implicit_dspu_template[8];
                                              /* reserved                      */
        unsigned char   implicit_ls_limit;    /* max number of implicit links  */
        unsigned char   reserv2;              /* reserved                      */
        unsigned char   implicit_dspu_services;
                                              /* implicit links support DSPUs  */
        unsigned char   implicit_deact_timer; /* Implicit link HPR link        */
                                              /* deactivation timer            */
        unsigned short  act_xid_exchange_limit;
                                              /* act.  XID exchange limit      */
        unsigned short  nonact_xid_exchange_limit;
                                              /* nonact.  XID exchange limit   */
        unsigned char   ls_xmit_rcv_cap;      /* LS transmit-receive           */
                                              /* capability                    */
        unsigned char   max_ifrm_rcvd;        /* max number of I-frames that   */
                                              /* can be received               */
        unsigned short  target_pacing_count;  /* Target pacing count           */
        unsigned short  max_send_btu_size;    /* Desired max send BTU size     */
        LINK_ADDRESS    dlc_data;             /* DLC data                      */
        LINK_ADDRESS    hpr_dlc_data;         /* HPR DLC data                  */
        unsigned char   implicit_cp_cp_sess_support;
                                              /* Implicit links allow CP-CP    */
                                              /* sessions                      */
        unsigned char   implicit_limited_resource;
                                              /* Implicit links are limited    */
                                              /* resource                      */
        unsigned char   implicit_hpr_support;
                                              /* Implicit links support HPR    */
        unsigned char   implicit_link_lvl_error;
                                              /* Implicit links support HPR    */
                                              /* link-level error recovery     */
        unsigned char   retired1;             /* reserved                      */
        TG_DEFINED_CHARS default_tg_chars;    /* Default TG chars              */
        unsigned char   discovery_support     /* Discovery function            */
                                              /* supported?                    */
        unsigned short  port_spec_data_len;   /* length of port spec data      */
        unsigned short  link_spec_data_len;   /* length of link spec data      */
} PORT_DEF_DATA;

typedef struct link_address
{
        unsigned short  length;               /* length                        */
        unsigned short  reserve1;             /* reserved                      */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                              /* address                       */
} LINK_ADDRESS;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_PORT

attributes

动词D t 性。此字段G一 v 位字段。 Z 一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + 此字段 h 置为 c，以指定在O f P v VCB D f >。

port_name

；定义端Z D { F。 | G一 v 在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须 h 置。

def_data.description

资源 5 w（返回 = QUERY_PORT O）。 | G一 v 在 > X I 显 > 字符集中D一 v 16 字Z字符串。 y P 16 v字Z都P效。

def_data.dlc_name

关 * DLC D { F， | G一 v 在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须 h 置。此 _ { DLC X须已 I 一 v DEFINE_DLC 动词定义。

def_data.port_type

指定端Z y 9 C线 7 D类型。值对& Z下P线 7 类型之一:

AP_PORT_NONSWITCHED
AP_PORT_SWITCHED
AP_PORT_SATF

注意，如果此字段 h 置为 AP_PORT_SATF，则 **ls_role** X须 h 置为 AP_LS_NEG。

def_data.port_attributes[0]

b G位字段。 | I 能 9 C值 AP_NO 或下P值:

AP_RESOLVE_BY_LINK_ADDRESS

| 指定一种分 f = 达呼 P D " T，方法G在 9 C P载在 S U XID3 O D CP { F（或Z c ID）之 0 先 9 C CONNECT_IN O D 4 7 X址 x 行分 f。 } 非字段 **port_type** h 置为 AP_PORT_SWITCHED，否则此位 + ；忽T。

def_data.implicit_uplink_to_en

v C Z BrNN: 指定如果相Z Z c 为端Z c，在此端Z下D隐 = 4 7 > G O行 4 7还G下行 4 7。 v 1 ；存在至相同伙 i D 4 7 1 E < G此字段D值，因为 1 存在 b 样D 4 7 1， + W先C其来确定 4 7 类型。

AP_NO

隐 = 4 7 为下行 4 7。

AP_YES

隐 = 4 7 为O行 4 7。

其 | Z c 类型: 此字段 + 忽T。

def_data.port_number

端Z号。

def_data.tot_link_act_lim

4 7激活限制总计。指定I同1处Z活动状，D激活4 7 > D最大}目。 | X须大Z H Z **inb_link_act_lim** 和 **out_link_act_lim** 字段D总和。如果 **port_type** h置为 AP_PORT_NONSWITCHED, **ls_role** h置为 AP_LS_NEG 或 AP_LS_SEC, 则此字段X须h置为 1。如果 **ls_role** h置为 AP_LS_PRI, 则此字段D范围X须大Z H Z 1-256。如果此端Z C Z AnyNet DLC, X须9 C 65535。

def_data.inb_link_act_lim

入> 4 7激活限制。指定在此端Z O为入>激活# t D 4 7 > D }目。因此, I同1处Z活动状，D v > 4 7 > D最大}目为 **def_data.tot_link_act_lim** - **def_data.inb_link_act_lim**。如果 **port_type** h置为 AP_PORT_NONSWITCHED, **ls_role** h置为 AP_LS_NEG 或 AP_LS_PRI, 则此字段X须h置为 0。如果 **port_type** h置为 AP_PORT_NONSWITCHED, **ls_role** h置为 AP_LS_SEC, 则此字段X须h置为 1 或 0。如果此端Z C Z AnyNet DLC, X须9 C 0。

def_data.out_link_act_lim

v > 4 7激活限制。指定在此端Z O为v >激活# t D 4 7 > D }目。因此, I同1处Z活动状，D入> 4 7 > D最大}目为 **def_data.tot_link_act_lim** - **def_data.out_link_act_lim**。如果 **port_type** h置为 AP_PORT_NONSWITCHED, **ls_role** h置为 AP_LS_NEG, 则此字段X须h置为 0。如果 **ls_role** h置为 AP_LS_PRI, 则此字段X须H Z **tot_link_act_lim**。如果 **port_type** h置为 AP_PORT_NONSWITCHED, **ls_role** h置为 AP_LS_SEC, 则此字段X须h置为 1 或 0。如果此端Z C Z AnyNet DLC, X须9 C 0。

def_data.ls_role

4 7 > G +。 I为I协L（AP_LS_NEG）、主（AP_LS_PRI）或次（AP_LS_SEC）。4 7 > G +确定I **tot_act_lim**、 **inb_link_act_lim** 和 **out_link_act_lim** 字段指定D值（如O y v）之间D关系。注意, 如果 **port_type** h置为 AP_PORT_SATF, 则 **ls_role** X须h置为 AP_LS_NEG。

def_data.retry_flags

此字段指定如果在 **def_data.retry_flags** D DEFINE_LS O h置 AP_INHERIT_RETRY j志, 4 7 > I自动重T激活D u件。 | G一v位字段, " I 9 C任何下P 4位 OR（或Y作）组合。

AP_RETRY_ON_START

在" T激活1, 如果未从远L Z c S U =响&, 则重T 4 7激活。如果在" T激活1, 基>端Z G非活动D, 则 APPN + " T激活 |。

AP_RETRY_ON_FAILURE

如果在活动和暂挂活动1 4 S ' \, 则重T 4 7激活。如果在" T激活1, 基>端Z发z故O, 则 APPN + " T激活 |。

AP_RETRY_ON_DISCONNECT

如果4 7 I远L Z c以} #方=停止, 则重T 4 7激活。

AP_DELAY_APPLICATION_RETRIES

I &C L序启动D 4 7激活重T（9 C START_LS 或k s = 4 7激活）+ 9 C **activation_delay_timer** x行w =。

AP_INHERIT_RETRY

} K此字段中D j 志指定D重T u 件外，那些在基 > 端Z 定义D **retry_flags** 字段中指定D重T u 件也 + 9 C。

def_data.max_activation_attempts

此字段 ; z z 效果，} 非在 **def_data.retry_flags** 中D DEFINE_LS 中至Y h 置一 v j 志， DEFINE_LS O D **def_data.max_activation_attempts** h 置为 AP_USE_DEFAULTS。

此字段指定 1 远L Z c 无响&或基 > 端Z ；活动 1， L序允许重T D次 }。 | | 括自动重T和 I & C L序 } 动D激活 " T = _。

如果达 = 此极限，则；再 x 行自动重T。C u 件 I STOP_LS、STOP_PORT、STOP_DLC 或一 v I 功激活 4 位。START_LS 或 OPEN_LU_SSCP_SEC_RQ z z 一 v % v 激活 " T， | 在激活 ' \ 1 ；再重T。

c m > '无限制'。值 AP_USE_DEFAULTS <致在 DEFINE_DLC O a 供D **max_activiation_attempts** D 9 C。

def_data.activation_delay_timer

此字段 ; z z 效果，} 非在 **def_data.retry_flags** 中D DEFINE_LS 中至Y h 置一 v j 志， DEFINE_LS O D **activiation_delay_timer** h 置为 AP_USE_DEFAULTS。

此字段指定在自动重T间L序H待D k }，以及如果在 **def_data.retry_flags** 中 h 置 AP_DELAY_APPLICATION_RETRIES 位， I & C L序 } 动D激活 " T间 D k }。

值 AP_USE_DEFAULTS <致在 DEFINE_DLC O a 供D **activiation_delay_timer** D 9 C。

如果指定 c，则L序 + 9 C缺！计 1 器 1 间 30 k。

def_data.implicit_dspu_template

指定 DSPU 模 e（C DEFINE_DSPU_TEMPLATE 动词定义），如果 > X Z c 要为在此端Z O激活D隐 = 4 7 a 供 PU 集中，则 + 此模 e C作定义。如果 1 4 7 激活 1，指定D模 e ；存在（或已达 = 5 例极限），则激活 + ' \。| G 一 v 在 > X I 显 > 字符集中D一 v 8 字Z字符串。y P 8 v 字Z都G P 效D，" R X须 h 置。

如果 **def_data.implicit_dspu_services** 字段未 h 置为 AP_PU_CONCENTRATION，则此字段 + # t 。

def_data.implicit_ls_limit

指定 I 同 1 处Z 活动状，D隐 = 4 7 > D最大 } 目， | 括动，4 7 和为 Discovery 激活D 4 7。0 值m > 无限制，AP_NO_IMPLICIT_LINKS 值m > ；允许隐 = 4 7。

def_data.implicit.dspu_services

指定 > X Z c 在此端Z O激活D隐 = 4 7 中 + a 供 x 下N PU D服务。| h 置为下P 值之一:

AP_DLUR

> X Z c + 向下N PU a 供 DLUR 服务（9 C在 DEFINE_DLUR_DEFAULTS 动词中配置D缺！ DLUS）。此 h 置 v 1 > X Z c G网 g Z c 1 P 效。

AP_PU_CONCENTRATION

> X Z c + 向下N PU a 供 PU 集中（ " + 代 f I 指定 DSPU 模 e 在 **def_data.implicit_dspu_template** 字段中指定D定义）。

AP_NONE

> X Z c ；向下N PU a 供任何服务。

def_data.implicit_deact_timer

\ 限资源M放计 1 器（ k ）。如果 **implicit_limited_resource** h 置为 AP_YES 或 AP_NO_SESSIONS，则如果在C计 1 器期间； P }] 在 4 7 O穿过， _ P HPR 能&D隐 = 4 7 + 自动M放， R无会话 9 C 4 7 。

如果 **implicit_limited_resource** h 置为 AP_INACTIVITY，则如果在C计 1 器期间； P }] 在 4 7 O穿过，隐 = 4 7 + 自动M放。

值为 0-1000 k 范围内D { } 。缺 ! 值为 10 k 。

如果指定 c ，则 + 9 C缺 ! 值 30。否则，最小值为 5。（如果 h 置 | 小D值，则指定D值 + ；忽T， " 9 C 5。）注意， } 非 **implicit_limited_resource** h 置为 AP_NO，否则此N } # t 。

def_data.act_xid_exchange_limit

激活 XID ；换限制。

def_data.nonact_xid_exchange_limit

非激活 XID ；换限制。

def_data.ls_xmit_rcv_cap

指定 4 7 > 发M/ S U能&。 I 为 + 向同 1 （AP_LS_TWS）（也 F 为 + 工或全 + 工），或 _ 为 + 向； f （AP_LS_TWA）（也 F 为 k + 工）。

def_data.max_ifrm_rcvd

> X 4 7 > 在确认发M 0 I S U = D最大 I-帧 } 目。范围为 1--127。

def_data.target_pacing_count

1 和 32 767 之间D } 字值（ | 括 1 和 32 767），指 > 此 TG O BIND D期望 w = 窗 Z 大小。 v 1 执行固定， S w = 1 ，此 } 字 E P 效。注意， v 人通信或通信服务器 1 0 ； 9 C此值。

def_data.max_send_btu_size

I 从 4 7 > 发MD最大 BTU 大小。此值C Z 协L I 在一对 4 7 > 之间发MD最大 BTU 大小。如果在 4 7 O ；支V隐 = HPR-能&，则此字段X须 h 置为一 v 大Z H Z 99 D值。如果在 4 7 O支V隐 = HPR-能&，则此字段X须 h 置为一 v 大Z H Z 768 D值。

def_data.dlc_data.length

端 Z X址 $ 度。

def_data.dlc_data.address

端 Z X址。

def_data.hpr_dlc_data.length

HPR 端 Z X址 $ 度。

def_data.hpr_dlc_data.address

HPR 端 Z X址。目 0 1 支V HPR 4 7 1 9 C。此字段指定 I v 人通信或通信服务器发MD信息，存放在 9 C此端 Z D 4 7 > O ；换 XID3 O X'61' X制向

？D X'80' 子字段中。 | I v人通信或通信服务器在发= DLC D ACTIVATE_PORT O传] 。对Z支V HPR 4 7 D端Z，某些 DLC I要 s n d C信息。

def_data.implicit_cp_cp_sess_support
指定对Z在此端Z下D隐= 4 7 > G否允许 CP-CP 会话（AP_YES 或 AP_NO）。

def_data.implicit_limited_resource
指定1 ；P会话9 C 4 7 1， G否M放在此端Z下D隐= 4 7 >。 | h置为下P值之一:

AP_NO
隐= 4 7 ；G \限资源， R ；会自动M放。

AP_YES or AP_NO_SESSIONS
隐= 4 7 G \限资源， " 1 ；P活动会话9 C | G 1 +自动M放。

AP_INACTIVITY
隐= 4 7 G \限资源， 1 ；P活动会话9 C | G 1，或1在指定1间内（ I **implicit_deact_timer** 字段指定）4 7 O ；P }] w动1，隐= 4 7 +自动M放。

def_data.implicit_hpr_support
指定G否&在隐= 4 7 O支V HPR（AP_YES 或 AP_NO）。

def_data.implicit_link_lvl_error
指定 HPR 通信？G否9 C 4 S c错误恢4在隐= 4 7 O发M（AP_YES 或 AP_NO）。注意，如果 **implicit_hpr_support** h置为 AP_NO，则此N } # t 。

def_data.default_tg_chars
TG X性（N阅Z39页D『DEFINE_COS』）。C Z在此端Z下D隐= 4 7 >，或指定 **use_default_tg_chars** D已定义4 7 >。

def_data.discovery_supported
指定G否在此端Z O执行 Discovery 功能（AP_YES 或 AP_NO）。

def_data.port_spec_data_len
在 ACTIVATE_PORT 信号下， ；| D X传] =端Z D }] D $度。 }] & ，S =基> a构中。

def_data.link_spec_data_len
此字段& <终h置为c 。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PORT_NAME

AP_INVALID_DLC_NAME
AP_INVALID_PORT_TYPE
AP_INVALID_BTU_SIZE
AP_INVALID_LS_ROLE
AP_INVALID_LINK_ACTIVE_LIMIT
AP_INVALID_MAX_IFRM_RCVD
AP_INVALID_DSPU_SERVICES
AP_HPR_NOT_SUPPORTED
AP_DLUR_NOT_SUPPORTED
AP_PU_CONC_NOT_SUPPORTED
AP_INVALID_TEMPLATE_NAME
AP_INVALID_RETRY_FLAGS
AP_INVALID_IMPLICIT_UPLINK

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_PORT_ACTIVE

AP_DUPLICATE_PORT_NUMBER
AP_CANT_MODIFY_WHEN_ACTIVE
AP_CANT_MODIFY_VISIBILITY
AP_INVALID_IMPLICIT_UPLINK

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_TP

DEFINE_TP 动词定义B务L序（TP）信息， b些信息I Z c Y作员h ） TP ，S管理器在处理来自伙i LU D=达，S 1 9 C。此动词也I C Z修D一v以0定义B务L序D一v或多v字段（ + ；能C Z修D v人通信或通信服务器定义D B务L序）。

VCB a构

```
typedef struct define_tp
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   attributes;        /* verb attributes               */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   tp_name[64];       /* TP name                       */
        TP_CHARS        tp_chars;          /* TP characteristics            */
} DEFINE_TP;

typedef struct tp_chars
{
        unsigned char   description[RD_LEN]
                                           /* resource description          */
        unsigned char   conv_type;         /* conversation type             */
        unsigned char   security_rqd;      /* security support              */
        unsigned char   sync_level;        /* synchronization level support */
        unsigned char   dynamic_load;      /* dynamic load                  */
        unsigned char   enabled;           /* is the TP enabled?            */
        unsigned char   pip_allowed;       /* program initialization        */
                                           /* parameters supported          */
        unsigned char   duplex_support;    /* duplex supported              */
        unsigned char   reserv3[9];        /* reserved                      */
        unsigned short  tp_instance_limit; /* limit on currently active TP  */
                                           /* instances                     */
        unsigned short  incoming_alloc_timeout;
                                           /* incoming allocation timeout   */
        unsigned short  rcv_alloc_timeout; /* receive allocation timeout    */
        unsigned short  tp_data_len;       /* TP data length                */
        TP_SPEC_DATA    tp_data;           /* TP data                       */
} TP_CHARS;

typedef struct tp_spec_data
{
        unsigned char   pathname[256];     /* path and TP name              */
        unsigned char   parameters[64];    /* parameters for TP             */
        unsigned char   queued;            /* queued TP                     */
        unsigned char   load_type;         /* type of load-DETACHED/CONSOLE */
        unsigned char   dynamic_load       /* dynamic loading of TP enabled */
        unsigned char   reserved[5];       /* reserved                      */
} TP_SPEC_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_TP

attributes

动词D t性。此字段G一v位字段。 Z一位 | 含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

ｊ 6 VCB D q = 。 + 此字段 h 置为 c，以指定在O f P v VCB D f > 。

tp_name

；定义B务L序（TP）D { F。 b G一v 64 字Z EBCDIC 字符串，"以EBCDIC U q R n d。注意，v人通信或通信服务器；检 i 此字段D字符集。

tp_chars.description

资源 5 w（返回 = QUERY_TP_DEFINITION 和 QUERY_TP O）。| G一v在 > X I 显 > 字符集中D一v 16 字Z字符串。y P 16 v字Z都P效。

tp_chars.conv_type

指定此B务L序支V D对话类型。

AP_BASIC
AP_MAPPED
AP_EITHER

tp_chars.security_rqd

指定G否需要对话 2 全性信息以启动B务L序（AP_NO 或 AP_YES）。

tp_chars.sync_level

指定B务L序 y 支V D同 = 级。

AP_NONE

B务L序支V同 = 级“无”（None）。

AP_CONFIRM_SYNC_LEVEL

B务L序支V同 = 级“确认”（Confirm）。

AP_EITHER

B务L序支V同 = 级“无”（None）或“确认”（Confirm）。

AP_SYNCPT_REQUIRED

B务L序支V同 = 级“同 = c”（Sync-point）。

AP_SYNCPT_NEGOTIABLE

B务L序支V同 = 级“无”（None）、“确认”（Confirm）或“同 = c”（Sync-point）。

tp_chars.dynamic_load

指定B务L序G否I动，装入（AP_YES 或 AP_NO）。

tp_chars.enabled

指定B务L序G否I I功，S（AP_YES 或 AP_NO）。缺！为 AP_NO。

tp_chars.pip_allowed

指定B务L序G否I S U L序u < 化N }（PIP）（AP_YES 或 AP_NO）。

tp_chars.duplex_support

指w B务L序G全 + 工或 k + 工。

AP_FULL_DUPLEX

指定B务L序G全 + 工。

AP_HALF_DUPLEX
指定B务L序G k + 工。

AP_EITHER_DUPLEX
指定B务L序I为k + 工或全+工。

tp_chars.tp_instance_limit
同1活动D B务L序5例}目D限制。 c值m > 无限制。

tp_chars.incoming_alloc_timeout
指定一v =达，S +排队H待 RECEIVE_ALLOCATE D k }。 c m > 无，1，因此+；确定X挂起。

tp_chars.rcv_alloc_timeout
指定 RECEIVE_ALLOCATE 动词+排队H待一v，S D k }。 c m > 无，1，因此+；确定X挂起。

tp_chars.tp_data_len
k 5现相关B务L序}] D $度。

tp_spec_data
，S管理器在启动B务L序1 9 C D信息。如需P关如何9 C D细Z， k N阅*v人通信M户机/服务器通信L序h计*中D “，S管理器”。

tp_chars.tp_data.pathname
指定7 6和B务L序{ F。

tp_chars.tp_data.parameters
指定B务L序D N }。

tp_chars.tp_data.queued
指定G否对B务L序排队。

tp_chars.tp_data.load_type
指定如何装入B务L序。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_SYSTEM_TP_CANT_BE_CHANGED

AP_INVALID_CONV_TYPE
AP_INVALID_SYNC_LEVEL
AP_INVALID_DYNAMIC_LOAD
AP_INVALID_ENABLED
AP_INVALID_PIP_ALLOWED
AP_INVALID_DUPLEX_SUPPORT

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_CANT_MODIFY_VISIBILITY

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

重定 e **的** ' **果**: ？ v 字段 D 重定义 " 即 z 效（例如， 1 B 务 L 序 D 下一 v 5 例启动 1 ）。 + G，对 **incoming_alloc_timeout** 和 **rcv_alloc_timeout** 字段 D | D ； 0 响任何已排队 D ， S 或 RECEIVE_ALLOCATES。

DELETE_ADJACENT_NODE

DELETE_ADJACENT_NODED > } Z c 目 < }] b 中 k 某一相 Z Z c O 资源关 * D d 入项。

要从目 < 中 > } Z c D X 制 c 及其 LU, + **num_of_lus** h 置为 c。如果 **num_of_lus** 为非 c，则此动词 + C Z 从目 < 中 > } Z c LU, 而 # V X 制 c 定义；d。

如因任何原因动词 ' \，则；> } 任何目 < 项。

VCB a 构

DELETE_ADJACENT_NODE 动词 | 含多 v ADJACENT_NODE_LU 2 G。ADJACENT_NODE_LU a 构，S 在 DELETE_ADJACENT_NODE a 构 D 最后。

```
typedef struct delete_adjacent_node
{
        unsigned short  opcode;          /* verb operation code        */
        unsigned char   reserv2;         /* reserved                   */
        unsigned char   format;          /* format                     */
        unsigned short  primary_rc;      /* primary return code        */
        unsigned long   secondary_rc;    /* secondary return code      */
        unsigned char   cp_name[17];     /* CP name                    */
        unsigned short  num_of_lus;      /* number of LUs              */
} DELETE_ADJACENT_NODE;

typedef struct adjacent_node_lu
{
        unsigned char   wildcard_lu;     /* wildcard LU name indicator */
        unsigned char   fqlu_name[17];   /* fully qualified LU name    */
        unsigned char   reserv1[6];      /* reserved                   */
} ADJACENT_NODE_LU;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_DELETE_ADJACENT_NODE

format

j 6 VCB D q =。+ 此字段 h 置为 c，以指定在 O f P v VCB D f >。

cp_name

相 Z LEN 端 Z c 中 X 制 c D 全限定 {。{ F $ 度为 17 v 字 Z，" 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。(? v { F D $ 度最多 I 为 8 v 字 Z，" ；含 6 入 U q。)

num_of_lus

要 > } LU D } 目。如要 > } { v Z c 定义，h 置 c。此 } 字 m > z 在 DELETE_ADJACENT_NODE VCB 之后 D 相 Z LU 2 G D } 目。

adjacent_node_lu.wildcard_lu

m w 指定 D LU { F G 否 G 一 v 通配符 { F（AP_YES 或 AP_NO)。

adjacent_node_lu.fqlu_name

要 > } D LU { F。如果此 { F ；G 全限定 D，则假 h 为 CP { F D 网 g ID。

DELETE_ADJACENT_NODE

{ F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d。 | I 一 v 或 = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ， S。（ ？ v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 入 U q。）

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_CP_NAME

AP_INVALID_LU_NAME

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_INVALID_CP_NAME

AP_INVALID_LU_NAME

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_CN

如果y P关*端Z 4位，DELETE_CN > }和M放，S网g X制 i D内存。DELETE_CN也 I C Z从，S网g中> }选定D端Z。要b样做， C户& + **num_ports** 字段h置为一v非c值， " a供要> }端Z D { F。

VCB a构

```
typedef struct delete_cn
{
        unsigned short  opcode;         /* verb operation code        */
        unsigned char   attributes;     /* verb attributes            */
        unsigned char   format;         /* format                     */
        unsigned short  primary_rc;     /* primary return code        */
        unsigned long   secondary_rc;   /* secondary return code      */
        unsigned char   fqcn_name[17];  /* name of connection network */
        unsigned char   reserv1;        /* reserved                   */
        unsigned short  num_ports;      /* number of ports to delete  */
        unsigned char   port_name[8] [8];
                                        /* names of ports to delete   */
} DELETE_CN;
```

提供N数

& C L序a供下P N }:

opcode

AP_DELETE_CN

attributes

动词D t性。此字段G一v位字段。 Z一位 | 含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + 此字段h置为c，以指定在O f P v VCB D f >。

fqcn_name

要> }，S网g D { F（ $度为 17 v字Z）。此{ F I = v A型 EBCDIC 字符串组I，中间以一v EBCDIC c，S， "以 EBCDIC U q R n d。(? v { F D $度最多I为 8 v字Z， " ；含6入U q。)

num_ports

要在，S网g O > }D端Z D }目。如要> } { v Z c， h置c。

port_name

要> }端Z D { F（如果 **num_ports** 非c）。 ? v端Z { F G一v在> X I显> 字符集中D一v 8字Z字符串。 y P 8 v字Z都G P效D， " R X须h置。如果 **num_ports** 字段为c，则此字段# t。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_CN_NAME

AP_INVALID_NUM_PORTS_SPECIFIED

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c 停止而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_COS

DELETE_COS > } 一 v 服务级 d 入项， } 非 | G SNA 定义 D 缺！服务级之一。

VCB a 构

```
typedef struct delete_cos
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   cos_name[8];    /* class-of-service name */
} DELETE_COS;
```

提供 N 数

& C L 序 a 供下 P N } :

opcode

AP_DELETE_COS

format

j 6 VCB D q = 。 + 此字段 h 置为 c ，以指定在 O f P v VCB D f > 。

cos_name

服务级 { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N } :

primary_rc

AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_COS_NAME_NOT_DEFD

AP_SNA_DEFD_COS_CANT_BE_DELETE

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_DLC

DELETE_DLC ＞ } k DLC（如果 | 4位）关 * D y P端Z、 4 7 ＞和，S网g传d组（TG）。 y P D DLC X制 i ； ＞ }， " M放内存。 Z c Y作员h ）返回一v响&，指定 DLC G否I功＞ }。

注意，如果一v P PU k之关 * D 4 7 ＞ ； ＞ }（因为 | k DLC 关 * ），则在此 PU O定义D y P LU 也 + ； ＞ }。

VCB a构

```
typedef struct delete_dlc
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   attributes;     /* verb attributes       */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   dlc_name[8];    /* name of DLC           */

} DELETE_DLC;
```

提供N数

&C L序a供下P N }:

opcode

AP_DELETE_DLC

attributes

动词D t性。此字段G一v位字段。 Z一位 | 含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + 此字段h置为c，以指定在O f P v VCB D f ＞。

dlc_name

要＞ } DLC D { F。 | G一v在＞X I显＞字符集中D一v 8 字Z字符串。y P 8 v字Z都G P效D， " R X须h置。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DLC_NAME

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_DLC_ACTIVE

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_DOWNSTREAM_LU

此动词ｖＪＣＺ通信服务器。

VCB　ａ构

```
typedef struct delete_downstream_lu
{
        unsigned short  opcode;          /* verb operation code        */
        unsigned char   attributes;      /* verb attributes            */
        unsigned char   format;          /* format                     */
        unsigned short  primary_rc;      /* primary return code        */
        unsigned long   secondary_rc;    /* secondary return code      */
        unsigned char   dslu_name[8];    /* Downstream LU name         */
} DELETE_DOWNSTREAM_LU;
```

提供Ｎ数

＆ＣＬ序ａ供下ＰＮ｝:

opcode

AP_DELETE_DOWNSTREAM_LU

attributes

动词Ｄｔ性。此字段Ｇ一ｖ位字段。Ｚ一位｜含要定义资源ＤＩ见性，＂ｋ下Ｐ之一对＆:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

Ｉ４位 OR（或Ｙ作）至此字段中Ｄ其｜值如下:

AP_DELAY_IF_REQUIRED

指定Ｉ **dslu_name** 指定Ｄ下Ｎ LU １０活动，此动词＆在Ｌ序中排队，直＝ LU Ｉ为非活动。在ｂ种ｉｖ下，１ LU Ｉ为非活动１动词处理完Ｉ。

format

ｊ６ VCB Ｄｑ＝。＋此字段ｈ置为ｃ，以指定在Ｏｆ Ｐｖ VCB Ｄｆ＞。

dslu_name

；＞｝下Ｎ LU Ｄ｛Ｆ。ｂＧ一ｖ ８字Ｚ字母｝字Ｄ Ａ型 EBCDIC 字符串（以一ｖ字母＊头），＂以 EBCDIC Ｕｑ Ｒｎｄ。

返回Ｎ数

如果动词Ｉ功执行，则Ｌ序返回下ＰＮ｝:

primary_rc

AP_OK

如果因Ｎ｝错误而９动词；能执行，则Ｌ序返回下ＰＮ｝:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_DSLU_ACTIVE
AP_DELAYED_VERB_PENDING

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_LU_NAME

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_DOWNSTREAM_LU_RANGE

此动词ｖＪＣＺ通信服务器。

例如，一ｖ基｛ ＬＵＮＭＥ ｋ ＮＡＵ 范围 1-4 组合在一起，＋＞｝ ＬＵ LUNME001、LUNME002、 LUNME003 和 LUNME004。一ｖ基｛Ｄ＄度小Ｚ ５ ｖ非ｎｄ字符＋９Ｃ ＬＵ ｛ ＦＤ＄度小Ｚ ８ ｖ字符。

此动词＞｝ｙＰ在范围中Ｄ ＬＵ。如果范围内Ｄ某一 ＬＵ ；存在，则动词＋继续下一存在Ｄ ＬＵ。ｖ１在指定范围内；Ｐ ＬＵ １动词＇＼。

VCB a构

```
typedef struct delete_downstream_lu_range
{
        unsigned short  opcode;           /* verb operation code        */
        unsigned char   attributes;       /* verb attributes            */
        unsigned char   format;           /* format                     */
        unsigned short  primary_rc;       /* primary return code        */
        unsigned long   secondary_rc;     /* secondary return code      */
        unsigned char   dslu_base_name[5];/* Downstream LU base name    */
        unsigned char   min_nau;          /* min NAU address in range   */
        unsigned char   max_nau;          /* max NAU address in range   */
} DELETE_DOWNSTREAM_LU_RANGE;
```

提供Ｎ数

＆ＣＬ序ａ供下ＰＮ｝:

opcode

AP_DELETE_DOWNSTREAM_LU_RANGE

attributes

动词Ｄｔ性。此字段Ｇ一ｖ位字段。Ｚ一位｜含要定义资源ＤＩ见性，＂ｋ下Ｐ之一对＆:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

ｊ６ VCB Ｄｑ＝。＋此字段ｈ置为ｃ，以指定在Ｏｆ Ｐｖ VCB Ｄｆ＞。

dslu_base_name

下Ｎ ＬＵ ｛ Ｆ范围Ｄ基｛。ｂＧ一ｖ ５ 字Ｚ字母｝字Ｄ Ａ 型 EBCDIC 字符串（以一ｖ字母＊头），＂以 EBCDIC ＵｑＲｎｄ。此基｛之后＝加 ３ ｖ Ａ 型 EBCDIC ｝字字符，ｍ＞在 NAU 范围内？ｖ LU NAU Ｘ址Ｄ．ｘ制值。

min_nau

范围内Ｄ最小 NAU Ｘ址。｜Ｉ为 １ 至 255（｜括 １ 和 255）。

max_nau

范围内Ｄ最大 NAU Ｘ址。｜Ｉ为 １ 至 255（｜括 １ 和 255）。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc
AP_OK

如果因N}错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_NAU_ADDRESS

AP_INVALID_LU_NAME

如果因状，错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

AP_INVALID_LU_NAME
AP_DSLU_ACTIVE
AP_DELAYED_VERB_PENDING

secondary_rc
AP_INVALID_LU_NAME

如果因Z c P未启动而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_DSPU_TEMPLATE

此动词ｖＪＣＺ通信服务器。

VCB　ａ构

格式 1

```
typedef struct delete_dspu_template
{
        unsigned short  opcode;              /* verb operation code       */
        unsigned char   attributes;          /* verb attributes           */
        unsigned char   format;              /* format                    */
        unsigned short  primary_rc;          /* primary return code       */
        unsigned long   secondary_rc;        /* secondary return code     */
        unsigned char   template_name[8];    /* name of template          */
        unsigned short  num_of_dslu_templates;
                                             /* Number of DSLU templates  */
        unsigned char   reserv1[10];         /* reserved                  */
} DELETE_DSPU_TEMPLATE;

typedef struct dslu_template
{
        unsigned char   min_nau;             /* min NAU address in range  */
        unsigned char   max_nau;             /* max NAU address in range  */
        unsigned char   allow_timeout;       /* Allow timeout of host LU? */
        unsigned char   delayed_logon;       /* Allow delayed logon to    */
                                             /* host LU                   */
        unsigned char   reserv1[8];          /* reserved                  */
        unsigned char   host_lu[8];          /* host LU or pool name      */
} DSLU_TEMPLATE;
```

VCB　ａ构

格式 0

```
typedef struct delete_dspu_template
{
        unsigned short  opcode;              /* verb operation code       */
        unsigned char   attributes;          /* verb attributes           */
        unsigned char   format;              /* format                    */
        unsigned short  primary_rc;          /* primary return code       */
        unsigned long   secondary_rc;        /* secondary return code     */
        unsigned char   template_name[8];    /* name of template          */
} DELETE_DSPU_TEMPLATE;
```

提供Ｎ数

＆ＣＬ序ａ供下ＰＮ｝:

opcode

AP_DEFINE_DSPU_TEMPLATE

attributes

动词Ｄｔ性。此字段Ｇ一ｖ位字段。Ｚ一位｜含要定义资源ＤＩ见性，＂ｋ下Ｐ之一对＆:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

DELETE_DSPU_TEMPLATE

format

ｊ 6 VCB D q = 。 + 此字段 h 置为 c，以指定在O f P v VCB D f > 。

template_name

DSPU 模 e D { F 。(| k 在 PORT_DEF_DATA D **implicit_dspu_template** 字段中指定D { F 相对&)。 | G 一 v 在 > X I 显 > 字符集中D 一 v 8 字 Z 字符串。 y P 8 v 字 Z 都G P 效D， " R X 须 h 置。

num_of_dslu_templates

z 在 DEFINE_DSPU_TEMPLATE VCB 之后D DSLU 模 e 2 G D } 目。 | I 为 0 至 255 (| 括 0 和 255)。DSLU 模 e 作为 2 G = 加 = DELETE_DSPU_TEMPLATE VCB D 末尾。

dslu_template.min_nau

范围内D 最小 NAU X 址。 | I 为 1 至 255 (| 括 1 和 255)。

dslu_template.max_nau

范围内D 最大 NAU X 址。 | I 为 1 至 255 (| 括 1 和 255)。

def_data.allow_timeout

此字段 # t 。

def_data.delayed_logon

此字段 # t 。

dslu_template.host_lu

此字段 # t 。

返回N数

如果动词 I 功执行，则 L 序返回下 P N } :

primary_rc

AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TEMPLATE_NAME

AP_INVALID_NAU_RANGE

如果因相关 START_NODE N } 未 h 置而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_FOCAL_POINT

DELETE_FOCAL_POINT 动词 I C Z > } 某一指定类型和类 p D 9 c。如需 P 关 9 c 类型 D | 多信息， k N 阅 Z 64 页 D 『DEFINE_FOCAL_POINT』。如果一 v 活动 9 c ; > }， | + ; 取消。要取消（任何类型 D）活动 9 c，指定类型 AP_ACTIVE。如 8 份 9 c 或隐 = 9 c 在；活动 1 ; > }（通过指定 AP_BACKUP 或 AP_IMPLICIT），则任何 P 关 Z | D 存储信息 + ; > }。

注意，DEFINE_FOCAL_POINT 动词也 I C Z 取消 1 0 活动 D 9 c。C 重 4 功能 G 为向下兼容性 # t D。

VCB a 构

```
typedef struct delete_focal_point
{
        unsigned short  opcode;         /* verb operation code          */
        unsigned char   reserv2;        /* reserved                     */
        unsigned char   format;         /* format                       */
        unsigned short  primary_rc;     /* primary return code          */
        unsigned long   secondary_rc;   /* secondary return code        */
        unsigned char   reserved;       /* reserved                     */
        unsigned char   ms_category[8]; /* management services category */
        unsigned char   type;           /* type of focal point          */
} DELETE_FOCAL_POINT;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_DELETE_FOCAL_POINT

format

j 6 VCB D q =。+ 此字段 h 置为 c，以指定在 O f P v VCB D f >。

ms_category

管理服务类 p。| I 为如在 SNA 管理服务中定义 D 管理服务类 p D、以 4 字 Z a 构定义 D 值（以 EBCDIC U q R n d）之一，或为一 v 8 字 Z 1134 EBCDIC 2 装定义类型 D { F。

type 指定 ; > } 9 c D 类型。I 能 D 类型 G:

AP_ACTIVE

1 0 活动 9 c（I 为任意类型）；取消。

AP_IMPLICIT

隐 = 定义 ; > }。如果 1 0 活动 9 c G 一 v 隐 = 9 c，则 | + ; 取消。

AP_BACKUP

8 份定义 ; > }。如果 1 0 活动 9 c G 一 v 8 份 9 c，则 | + ; 取消。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_TYPE

AP_INVALID_CATEGORY_NAME

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因Z c 停止而 9 动词；能执行，则L 序返回下 P N } :

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_INTERNAL_PU

DELETE_INTERNAL_PU 动词 k s > } 某一 I DLUR a 供服务 D > X PU。 v 1 PU ; P 活动 SSCP-PU 会话 1 , 此动词会 I 功。

任何 k PU 关 * D LU + ; > } 。

VCB a 构

```
typedef struct delete_internal_pu
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   attributes;     /* verb attributes       */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   pu_name[8];     /* internal PU name      */
} DELETE_INTERNAL_PU;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_DELETE_INTERNAL_PU

attributes

动词D t 性。此字段G一 v 位字段。 Z 一位 | 含要定义资源D I 见性, " k 下 P 之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q = 。 + 此字段 h 置为 c , 以指定在O f P v VCB D f > 。

pu_name

; > } 内 ? PU D { F 。 b G一 v 8 字 Z 字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）, " 以 EBCDIC U q R n d 。

返回N数

如果动词 I 功执行, 则 L 序返回下 P N } :

primary_rc

AP_OK

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PU_NAME

AP_INVALID_PU_TYPE

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PU_NOT_RESET

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_LOCAL_LU

DELETE_LOCAL_LU k s > } > X LU 定义。

VCB a 构

```
typedef struct delete_local_lu
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   lu_name[8];     /* local LU name         */
} DELETE_LOCAL_LU;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_DELETE_LOCAL_LU

format

j 6 VCB D q = 。 + 此字段 h 置为 c , 以指定在O f P v VCB D f > 。

lu_name

; 定义 > X LU D { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）, " 以 EBCDIC U q R n d 。

返回N数

如果动词 I 功执行, 则 L 序返回下 P N } :

primary_rc

AP_OK

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_CANT_DELETE_CP_LU

如果因 Z c P 未启动而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_LS

DELETE_LS 检 i 4 7 > 已在以 0 定义和 4 位。 | > } 4 7 > X制 i , " 从 Z c Y作员 h) 返回一 v 响&, 指定 4 7 > G否已 I 功 > } 。注意, 任何在 PU O定义D、9C4 7 > D LU 也 + ; > } 。

VCB a 构

```
typedef struct delete_ls
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   attributes;     /* verb attributes       */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   ls_name[8];     /* name of link station  */
} DELETE_LS;
```

提供N数

&C L 序 a 供下 P N } :

opcode

AP_DELETE_LS

attributes

动词D t 性。此字段G一 v 位字段。 Z 一位 | 含要定义资源D I 见性, " k 下 P 之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q = 。 + 此字段 h 置为 c , 以指定在O f P v VCB D f > 。

ls_name

; > } 4 7 > D { F 。 | G一 v 在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D, " R X须 h 置。

返回N数

如果动词 I 功执行, 则L 序返回下 P N } :

primary_rc

AP_OK

如果因N } 错误而 9 动词 ; 能执行, 则L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LINK_NAME

如果因状 , 错误而 9 动词 ; 能执行, 则L 序返回下 P N } :

primary_rc

AP_STATE_CHECK

secondary_rc

AP_LS_ACTIVE

AP_INVALID_LINK_NAME

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_LU_0_TO_3

此动词C Z > } 一 v X定D LU。

VCB a 构

```
typedef struct delete_lu_0_to_3
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   attributes;     /* verb attributes       */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   lu_name[8];     /* LU name               */
} DELETE_LU_0_TO_3;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_DELETE_LU_0_TO_3

attributes

动词D t 性。此字段G一 v 位字段。 Z 一位 | 含要定义资源D I 见性， " k 下 P 之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q = 。 + 此字段 h 置为 c ，以指定在O f P v VCB D f > 。

lu_name

要 > } LU D { F 。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。

返回N数

如果动词 I 功执行，则L 序返回下 P N } :

primary_rc

AP_OK

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_CANT_DELETE_IMPLICIT_LU

如果因状，错误而 9 动词；能执行，则L 序返回下 P N } :

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_LU_NAME

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_LU_0_TO_3_RANGE

此动词C Z > } 某一范围内D LU。Z c Y作员 a 供一 v 基 { 和一 v NAU 范围。LU { F I + 基 { 和 NAU X址组合在一起 z I 。

例如，一 v 基 { LUNME k NAU 范围 1-4 组合在一起，+ > } LU LUNME001、LUNME002、LUNME003 和 LUNME004。一 v 基 { D $ 度小Z 5 v 非 n d 字符 + 9 C LU { F D $ 度小Z 8 v 字符。

在范围内D y P LU 都 + ; > } 。如果范围内D某一 LU ；存在，则动词 + 继续下一存在D LU。1 在指定范围内；P LU 1 动词 + ' \ 。

VCB a 构

格式 1

```
typedef struct delete_lu_0_to_3_range
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   base_name[6];           /* base name                  */
        unsigned char   min_nau;                /* minimum NAU address        */
        unsigned char   max_nau;                /* maximum NAU address        */
        unsigned char   name_attributes;        /* Attributes of base_name    */
        unsigned char   base_number;            /* Base number for LU names   */
        unsigned char   reserv5[16];               /* reserved                    */
} DELETE_LU_0_TO_3_RANGE;
```

VCB a 构

格式 0

```
typedef struct delete_lu_0_to_3_range
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   base_name[5];           /* base name                  */
        unsigned char   min_nau;                /* minimum NAU address        */
        unsigned char   max_nau;                /* maximum NAU address        */
        unsigned char   reserv3;                /* reserved                   */
} DELETE_LU_0_TO_3_RANGE;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_DELETE_LU_0_TO_3_RANGE

attributes

动词D t 性。此字段G一 v 位字段。Z 一位 | 含要定义资源D I 见性，" k 下 P 之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format
j 6 VCB D q = 。 + 此字段 h 置为 c，以指定在 O f P v VCB D f > 。

base_name
基 > LU { F。 b G 一 v 5 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d。此基 { 之后 = 加 3 v A 型 EBCDIC } 字字符，m > 在 NAU 范围内？v LU NAU X 址 D . x 制值。

min_nau
范围内 D 最小 NAU X 址。| I 为 1 至 255（| 括 1 和 255）。

max_nau
范围内 D 最大 NAU X 址。| I 为 1 至 255（| 括 1 和 255）。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N }：

primary_rc
AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_NAU_ADDRESS

AP_INVALID_LU_NAME

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc
AP_STATE_CHECK

secondary_rc
AP_INVALID_LU_NAME

AP_CANT_DELETE_IMPLICIT_LU

如果因系统；P ("（ _ P 从 t LU 支 V）而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc
AP_INVALID_VERB

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }：

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_LU_POOL

此动词C Z > } 一 v LU X或从X中> } LU。如未指定 LU { F，则+ > } { v X。1 LU X中指定 LU 或 LU X > m；再存在1，此动词I 功完I。 v 1无指定D LU存在、或在指定X中无 LU 1，动词会 ' \。

VCB a 构

```
typedef struct delete_lu_pool
{
        unsigned short  opcode;                 /* verb operation code        */
        unsigned char   attributes;             /* verb attributes            */
        unsigned char   format;                 /* format                     */
        unsigned short  primary_rc;             /* primary return code        */
        unsigned long   secondary_rc;           /* secondary return code      */
        unsigned char   pool_name[8];           /* LU pool name               */
        unsigned short  num_lus;                /* number of LUs to add       */
        unsigned char   lu_names[10][8];        /* LU names                   */
} DELETE_LU_POOL;
```

提供N数

&C L序a 供下P N }:

opcode

AP_DELETE_LU_POOL

attributes

动词D t 性。此字段G一v 位字段。 Z 一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + 此字段h 置为c，以指定在O f P v VCB D f >。

pool_name

LU X D { F。 y P 8 v字Z都G P效D， " R X须h 置。此{ F G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v 字母 * 头）， " 以 EBCDIC U q R n d。

num_lus

在 **lu_names** P m中指定D LU } 目。

lu_names

要> } LU D { F。 ? v { F G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v 字母 * 头）， " 以 EBCDIC U q R n d。

返回N数

如果动词I 功执行，则L 序返回下P N }:

primary_rc

AP_OK

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_POOL_NAME

AP_INVALID_LU_NAME
AP_INVALID_NUM_LUS

如果因系统；P (" （ _ P 从 t LU 支V ）而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_INVALID_VERB

如果因 Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_MODE

DELETE_MODE k s > } 方 = 定义。CPSVCMG、SNASVCMG 和其 | j 准 SNA 方 = D缺！定义 + ；会；> }。

VCB a 构

```
typedef struct delete_mode
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   mode_name[8];   /* mode name             */
} DELETE_MODE;
```

提供N数

&C L 序 a 供下 P N }:

opcode

AP_DELETE_MODE

format

j 6 VCB D q =。+ 此字段 h 置为 c，以指定在O f P v VCB D f >。

mode_name

方 = { F。b G一 v 8 字 Z 字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头），"以 EBCDIC U q R n d。

返回N数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_CP_OR_SNA_SVCMG_UNDELETABLE

AP_MODE_UNDELETABLE
AP_DEL_MODE_DEFAULT_SPCD
AP_MODE_NAME_NOT_DEFD

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

DELETE_MODE

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_PARTNER_LU

DELETE_PARTNER_LU k s > } 伙 i LU 定义。

VCB a 构

```
typedef struct delete_partner_lu
{
        unsigned short  opcode;         /* verb operation code     */
        unsigned char   reserv2;        /* reserved                */
        unsigned char   format;         /* format                  */
        unsigned short  primary_rc;     /* primary return code     */
        unsigned long   secondary_rc;   /* secondary return code   */
        unsigned char   fqplu_name[17]; /* fully qualified partner */
                                        /* LU name                 */
} DELETE_PARTNER_LU;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_DELETE_PARTNER_LU

format

j 6 VCB D q = 。 + 此字段 h 置为 c , 以指定在O f P v VCB D f > 。

fqplu_name

伙 i LU D全限定 { F 。 { F $ 度为 17 v字Z , " 以 EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。(? v { F D $ 度最多 I 为 8 v字Z , " ; 含 6 入U q 。)

返回N数

如果动词 I 功执行, 则 L 序返回下 P N } :

primary_rc

AP_OK

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PLU_NAME

如果因 Z c P 未启动而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_PORT

DELETE_PORT > } k端Z（如果 | 4位）关 * D y P 4 7 > 和，S网g传d组（TG）。然后 | > } 端Z D X制 i， M放内存， " 从Z c Y作员h) 返回一v响&，指 > 端Z G否已 I 功 > } 。

注意，如果一v P PU k之关 * D 4 7 > ；> }（因为 | k端Z关 * ），则在此 PU O定义D y P LU 也 + ；> } 。

VCB a 构

```
typedef struct delete_port
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   attributes;     /* verb attributes       */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   port_name[8];   /* name of port          */
} DELETE_PORT;
```

提供N数

&C L序a供下P N }:

opcode

AP_DELETE_PORT

attributes

动词D t性。此字段G一v位字段。 Z一位 | 含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + 此字段h置为c，以指定在O f P v VCB D f > 。

port_name

；> } 端Z D { F。 | G一v在 > X I 显 > 字符集中D一v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须h置。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因N } 错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PORT_NAME

如果因状，错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PORT_ACTIVE

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_TP

DELETE_TP k s > } B务L序（TP）定义。

VCB a构

```
typedef struct delete_tp
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   attributes;     /* verb attributes        */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   tp_name[64];    /* TP name               */
} DELETE_TP;
```

提供N数

&C L序a供下P N }:

opcode

AP_DELETE_TP q = j 6 VCB D q =。+此字段h置为c，以指定在O f P v VCB D f >。

attributes

动词D t性。此字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

tp_name

B务L序D{ F。L序；检i此字段D字符集。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TP_NAME

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因Z c停止而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

第5章 $活和释放动词

>BZhvKCZ激活和M放D动词:

- }]47X制（DLC）
- 内？ PU
- 端Z
- 47>
- 会话
- 对话组

>BZ还hvKCZks支V_性能7I选择（HPR）D，SD76*关D动词。

© Copyright IBM Corp. 1989, 1997, 1998

START_DLC

START_DLC k s }] 4 7 X制（DLC）D激活。 | f后返回指> 5 w DLC DM放G否I功。注意即9 ; P为 DLC 定义D端Z，DLC 仍能够启动。如需P关 DLC、端Z和4 7 >间关系D | 多信息， k N阅 Z16页D『DLC 过L、端Z和4 7 >』 。

VCB a构

```
typedef struct start_dlc
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   dlc_name[8];    /* name of DLC           */
} START_DLC;
```

提供N数

& C L序a供下P N }:

opcode

AP_START_DLC

format

j 6 VCB D q =。+ C字段h置为c，以指定O f P v D VCB D f >。

dlc_name

要启动D }] 4 7 X制5例D { 字。 b G v以I显> D > X字符集m > D 8 字Z D字符串， C字符串X须已I DEFINE_DLC 动词定义。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_DLC

如果C动词因 DLC }在M放而未能执行，则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_DLC_DEACTIVATING

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

START_INTERNAL_PU

START_INTERNAL_PU 动词k s从t LU k s L序（DLUR）u < 化过去定义D I DLUR 服务D > X PU D SSCP-PU 会话M放。

VCB a构

```
typedef struct start_internal_pu
{
        unsigned short  opcode;             /* verb operation code   */
        unsigned char   reserv2;            /* reserved              */
        unsigned char   format;             /* format                */
        unsigned short  primary_rc;         /* primary return code   */
        unsigned long   secondary_rc;       /* secondary return code */
        unsigned char   pu_name[8];         /* internal PU name      */
        unsigned char   dlus_name[17];      /* DLUS name             */
        unsigned char   bkup_dlus_name[17]; /* Backup DLUS name      */
} START_INTERNAL_PU;
```

提供N数

&C L序a供下P N }:

opcode

AP_START_INTERNAL_PU

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

pu_name

+ k s SSCP-PU 会话启动w ? D内? PU D { 字。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * <），I EBCDIC U q R n d。

dlus_name

为x定 PU D k s SSCP-PU 会话激活，DLUR + k之 * 系D从t LU 服务器（DLUS）Z c D { 字。C值&置为全c，或I = v A 型 EBCDIC 字符串组I、中间以一v EBCDIC c ，S、 " 以 EBCDIC U q R n d D 17 字Z字符串。（ ? v { F D $ 度最多I为 8 v字Z， R ；含6入U q。）C值t性置换K在 DEFINE_INTERNAL_PU 动词中指定D值。如果C字段置为全c，则 + 会9 C在 DEFINE_INTERNAL_PU 动词中指定D DLUS 。如果；P DLUS 已在 DEFINE_INTERNAL_PU 动词中指定，那4 + 会9 C全V缺！值（如果I DEFINE_DLUR_DEFAULTS 动词指定）。

bkup_dlus_name

对Z x定 PU，DLUR + 会作为8份 DLUS 储存D DLUS Z c D { 字。C值&置为全c，或I = v A 型 EBCDIC 字符串组I、中间以一v EBCDIC c ，S、 " 以 EBCDIC U q R n d D 17 字Z字符串。（ ? v { F D $ 度最多I为 8 v字Z， R ；含6入U q。）C值t性置换K在 DEFINE_INTERNAL_PU 动词中指定D值。如果C字段置为全c，则I DEFINE_INTERNAL_PU 动词指定D 8份 DLUS {，+会作为C PU D 8份 DLUS # t。如果；P 8份 DLUS I DEFINE_INTERNAL_PU 动词指定，全V 8份缺！ DLUS（如果I DEFINE_DLUR_DEFAULTS 动词定义）作为C PU D缺！8份# t。

返回N数

如果C动词I功执行，则L序返回下PN}:

primary_rc
AP_OK

如果C动词因N}错误而未能执行，则L序返回下PN}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DLUS_NAME

AP_INVALID_BKUP_DLUS_NAME

如果C动词因状，错误而未能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_NO_DEFAULT_DLUS_DEFINED

AP_PU_NOT_DEFINED
AP_PU_ALREADY_ACTIVATING
AP_PU_ALREADY_ACTIVE

如果动词未I功执行，则L序返回下PN}:

primary_rc
AP_UNSUCCESSFUL

secondary_rc
AP_DLUS_REJECTED

AP_DLUS_CAPS_MISMATCH
AP_PU_FAILED_ACTPU

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

START_LS

START_LS k s 4 7 D激活。 | 作为&答返回，指定 4 7 G否 I 功激活。

如需 P 关 DLC、端 Z 和 4 7 > 间关系 D | 多信息， k N阅 Z 16页 D 『DLC 过 L 、端 Z 和 4 7 > 』 。

VCB a 构

```
typedef struct start_ls
{
        unsigned short  opcode;         /* verb operation code        */
        unsigned char   reserv2;        /* reserved                   */
        unsigned char   format;         /* format                     */
        unsigned short  primary_rc;     /* primary return code        */
        unsigned long   secondary_rc;   /* secondary return code      */
        unsigned char   ls_name[8];     /* name of link station       */
        unsigned char   enable;         /* whether the link is enabled*/
        unsigned char   reserv3[3];     /* reserved                   */
} START_LS;
```

提供N数

& C L 序 a 供下 P N } :

opcode

AP_START_LS

format

j 6 VCB D q = 。 + C字段 h 置为 c ，以指定O f P v D VCB D f > 。

ls_name

要启动D 4 7 > { 。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v 字 Z 都G P 效D， R X须 h 置。**ls_name** D值X须 k DEFINE_LS 动词匹配。

enable

为启动 4 7 h 置C字段。如果C字段置为 AP_ACTIVATE，那 4 4 7 启动。否则， C 4 7 ；会启动， R下 P 值G ； I 能D。 b 些值能够一起作或运 c 。

AP_AUTO_ACT

在 > X Z c k s 1，能够 f 后启动D 4 7 。如果在 DEFINE_LS 动词中 **auto_act_supp** 置为 AP_YES ，则C值 v 在此 1 P 效。

AP_REMOTE_ACT

能够 f 后 I 远 L Z c 激活D 4 7 。 b ；D d **disable_remote_act** D定义值。

返回N数

如果C动词 I 功执行，则 L 序返回下 P N } :

primary_rc

AP_OK

如果C动词因 N } 错误而未能执行，则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LINK_NAME_SPECIFIED

如果C动词因状，错误而未能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PORT_INACTIVE

AP_ACTIVATION_LIMITS_REACHED
AP_PARALLEL_TGS_NOT_SUPPORTED
AP_ALREADY_STARTING
AP_LINK_DEACT_IN_PROGRESS

如果C动词因在4 7 d为活动之0 I f后D STOP_LS 或 STOP_PORT 动词放弃，则L序返回下PN}:

primary_rc
AP_CANCELLED

secondary_rc
AP_LINK_DEACTIVATED

如果C动词因；能通过4 7软件R = 伙 i 而未能执行，则L序返回下PN}:

primary_rc
AP_LS_FAILURE

secondary_rc
AP_PARTNER_NOT_FOUND

如果C动词因}1 4 7 (" 1发z 4 7错误，则L序返回下PN}:

primary_rc
AP_LS_FAILURE

secondary_rc
AP_ERROR

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

START_PORT

START_PORT k s端Z D启动。 | 返回指> 5 w端Z G否I 功启动。即9 ; P 4 7 >已为其定义， C端Z 仍能够启动; + G如果 | D 8代 DLC G ；活动D，那4 | + ；会启动。

如需P关 DLC、端Z 和4 7 >间关系D | 多信息， k N阅Z 16页D『DLC 过L、端Z 和4 7 > 』 。

VCB a 构

```
typedef struct start_port
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   port_name[8];   /* name of port          */
} START_PORT;
```

提供N数

& C L序a 供下P N }:

opcode

AP_START_PORT

format

j 6 VCB D q =。 + C字段h 置为c，以指定O f P v D VCB D f >。

port_name

要启动D端Z { 。 b G v以I 显> D > X字符集m > D 8 字Z D字符串 " R X须k DEFINE_PORT 端Z 匹配。

返回N数

如果C 动词I 功执行，则L 序返回下P N }:

primary_rc

AP_OK

如果C 动词因N } 错误而未能执行，则L 序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PORT_NAME

如果C 动词因状，错误而未能执行，则L 序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_DLC_INACTIVE

AP_STOP_PORT_PENDING
AP_DUPLICATE_PORT

如果C动词因已放弃而未能执行，则L序返回下PN}:

primary_rc
AP_CANCELLED

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

STOP_DLC

STOP_DLC k s让 DLC 停止。 | 返回指 > 5 w DLC G否 I 功停止。为停止C DLC D端 Z O任何 4 7 > 自动重 T 激活，STOP_DLC 同样也 C Z 指 > L 序。

VCB a 构

```
typedef struct stop_dlc
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   stop_type;      /* stop type             */
        unsigned char   dlc_name[8];    /* name of DLC           */
} STOP_DLC;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_STOP_DLC

format

j 6 VCB D q =。 + C字段 h 置为 c，以指定O f P v D VCB D f >。

stop_type

DLC & C停止D类型。

AP_ORDERLY_STOP

在停止端 Z 之 0 & C 执行 e } Y作D Z c。

AP_IMMEDIATE_STOP

& C " 即停止 DLC D Z c。

dlc_name

要停止D DLC {。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串 " R X须 k DEFINE_DLC 动词匹配。

返回N数

如果C动词 I 功执行，则L序返回下 P N }:

primary_rc

AP_OK

如果C动词因N } 错误而未能执行，则L序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_DLC

AP_UNRECOGNIZED_DEACT_TYPE

如果C动词因状，错误而未能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_STOP_DLC_PENDING

如果C动词因已放弃而未能执行，则L序返回下PN}:

primary_rc
AP_CANCELLED

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

STOP_INTERNAL_PU

STOP_INTERNAL_PU 动词k s 从t LU k s L序（DLUR）u <化过去定义D I DLUR 服务D > X PU D SSCP-PU 会话M放。

VCB a构

```
typedef struct stop_internal_pu
{
        unsigned short  opcode;         /* verb operation code    */
        unsigned char   reserv2;        /* reserved               */
        unsigned char   format;         /* format                 */
        unsigned short  primary_rc;     /* primary return code    */
        unsigned long   secondary_rc;   /* secondary return code  */
        unsigned char   pu_name[8];     /* internal PU name       */
        unsigned char   stop_type;      /* type of stop requested */
} STOP_INTERNAL_PU;
```

提供N数

& C L序a供下P N }:

opcode

AP_STOP_INTERNAL_PU

format

j 6 VCB D q =。+ C字段h置为c，以指定O f P v D VCB D f >。

pu_name

+ M放D SSCP-PU 会话D内？ PU D { 字。b G v 8 字Z D字母 } 字集D A型 EBCDIC 字符串（以一v字母 * <），I EBCDIC U q R n d。

stop_type

指定为 PU k s D停止类型。在启动 SSCP-PU 会话0， P次序D停止+ M放y P基> PLU-SLU 和 SSCP-LU 会话。

AP_ORDERLY_STOP
AP_IMMEDIATE_STOP

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_STOP_TYPE

如果C动词因状，错误而未能执行，则L序返回下P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PU_NOT_DEFINED

AP_PU_ALREADY_DEACTIVATING
AP_PU_NOT_ACTIVE

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

STOP_LS

STOP_LS k s 4 7 > D激活。 | D返回指定端Z G否I功停止。STOP_LS 同样能C Z { 止4 7 > D远L启动或{ 止4 7 > k s D启动。STOP_PORT 同样也C Z指> L序为停止在端Z自动重T任何4 7 > D激活。

VCB a构

```
typedef struct stop_ls
{
        unsigned short  opcode;         /* verb operation code          */
        unsigned char   reserv2;        /* reserved                     */
        unsigned char   format;         /* format                       */
        unsigned short  primary_rc;     /* primary return code          */
        unsigned long   secondary_rc;   /* secondary return code        */
        unsigned char   stop_type;      /* stop type                    */
        unsigned char   ls_name[8];     /* name of link station         */
        unsigned char   disable;        /* whether the link is disabled */
        unsigned char   reserved[3];    /* reserved                     */
} STOP_LS;
```

提供N数

&C L序a供下P N }:

opcode

AP_STOP_LS

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

stop_type

端Z & C停止D种类。

AP_ORDERLY_STOP

在停止端Z之0 & C执行e } Y作D Z c。

AP_IMMEDIATE_STOP

& C "即停止4 7 > D Z c。

ls_name

要停止D 4 7 > {。 b G v以I显> D > X字符集m > D 8字Z D字符串。 y P D 8 v字Z都G P效D， R X须h置。**ls_name** D值X须k DEFINE_LS 动词匹配。

disable

C动词m w G否{ C远L启动或C 4 7 > k s启动。如果置为 AP_NO，则4 7 > 返回 = 来自 DEFINE_LS 动词D **auto_act_supp** 和 **disable_remote_act** H值x h D状，。否则，下P值P I能（ " R能够一起做或运c）。

AP_AUTO_ACT

在> X Z c k s 1， C 4 7 ；能重新激活。

AP_REMOTE_ACT

C 4 7 ；能I远L Z c激活。对Z配置**disable_remote_act** 为AP_YES D 4 7，忽T C位（ I远L Z c激活D总I STOP_LS { C）。

如果 **disable** 字段；P 置为 AP_NO ， 那 4 为K h 置 **disable** 字段D 目D， STOP_LS 能够为；活动D或 } 在 x 行M放处理D 4 7 发 v 。

返回N数

如果C 动词 I 功执行，则L 序返回下 P N } :

primary_rc
AP_OK

如果C 动词因N } 错误而未能执行，则L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_UNRECOGNIZED_DEACT_TYPE

AP_LINK_NOT_DEFD

如果C 动词因状，错误而未能执行，则L 序返回下 P N } :

primary_rc
AP_STATE_CHECK

secondary_rc
AP_LINK_DEACT_IN_PROGRESS

如果C 动词因已放弃而未能执行，则L 序返回下 P N } :

primary_rc
AP_CANCELLED

如果C 动词因 Z c P 未启动而未能执行，则L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果C 动词因 Z c 停止而未能执行，则L 序返回下 P N } :

primary_rc
AP_NODE_STOPPING

如果C 动词因系统错误而未能执行，则L 序返回下 P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

STOP_PORT

STOP_PORT k s停止端Z。 | D返回指定端Z G否I功停止。STOP_PORT 同样也C Z指> L序为停止在端Z自动重T任何4 7 > D激活。

VCB a构

```
typedef struct stop_port
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   stop_type;      /* Stop Type             */
        unsigned char   port_name[8];   /* name of port          */
} STOP_PORT;
```

提供N数

&C L序a供下P N }:

opcode

AP_STOP_PORT

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

stop_type

端Z & C停止D种类。

AP_ORDERLY_STOP

在停止端Z之0 & C执行e } Y作D Z c。

AP_IMMEDIATE_STOP

& C " 即停止端Z D Z c。

port_name

要停止D端Z { 。 b G v以I显> D > X字符集m > D 8 字Z D字符串 " R X须k DEFINE_PORT 动词匹配。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PORT_NAME

AP_UNRECOGNIZED_DEACT_TYPE

如果C动词因状，错误而未能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_STOP_PORT_PENDING

如果C动词因已放弃而未能执行，则L序返回下PN}:

primary_rc
AP_CANCELLED

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

ACTIVATE_SESSION

ACTIVATE_SESSION 动词k s在> X LU 和指定D 9 C X定模= 字符WD伙i LU 间D会话D激活。

VCB a构

Format 1

```
typedef struct activate_session
{
        unsigned short  opcode;         /* verb operation code     */
        unsigned char   reserv2;        /* reserved                */
        unsigned char   format;         /* format                  */
        unsigned short  primary_rc;     /* primary return code     */
        unsigned long   secondary_rc;   /* secondary return code   */
        unsigned char   lu_name[8];     /* local LU name           */
        unsigned char   lu_alias[8];    /* local LU alias          */
        unsigned char   plu_alias[8];   /* partner LU alias        */
        unsigned char   mode_name[8];   /* mode name               */
        unsigned char   fqplu_name[17]; /* fully qualified partner */
                                        /* LU name                 */
        unsigned char   polarity;       /* requested session       */
                                        /* polarity                */
        unsigned char   session_id[8];  /* session identifier      */
        unsigned char   cnos_permitted; /* is implicit CNOS        */
                                        /* permitted?              */*/
        unsigned char   reserv4[15];    /* reserved                */
} ACTIVATE_SESSION;
```

Format 0 (back-level)

```
typedef struct activate_session
{
        unsigned short  opcode;         /* verb operation code     */
        unsigned char   reserv2;        /* reserved                */
        unsigned char   format;         /* format                  */
        unsigned short  primary_rc;     /* primary return code     */
        unsigned long   secondary_rc;   /* secondary return code   */
        unsigned char   lu_name[8];     /* local LU name           */
        unsigned char   lu_alias[8];    /* local LU alias          */
        unsigned char   plu_alias[8];   /* partner LU alias        */
        unsigned char   mode_name[8];   /* mode name               */
        unsigned char   fqplu_name[17]; /* fully qualified partner */
                                        /* LU name                 */
        unsigned char   polarity;       /* requested session       */
                                        /* polarity                */
        unsigned char   session_id[8];  /* session identifier      */
} ACTIVATE_SESSION;
```

提供N数

& C L序a供下P N }:

opcode

AP_ACTIVATE_SESSION

format

j 6 VCB D q =。+ C字段h置为c，以指定O f P v D VCB D f >。

lu_name

> X LU k s C Z M放会话D LU D { 字。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段置为全c，则 **lu_alias** 字段 + C Z v 定 > X LU。

lu_alias

> X LU k s C Z M放会话D p { 。b G v 以 I 显 > D > X字符集m > D 8 字 Z D字符串。v 1 **lu_name** 字段h置为全c 1，| E P效，在b种 i v下。y P 8 v字Z都G P效D " R X须h置。如果**lu_alias** 和 **lu_name** = _ 置为全 c，则C动词转发x k X制c相关 * D LU（缺！ LU)。

plu_alias

为 > X LU y知D伙 i LU D p { 。b v { 字X须k在配置过L中 (" D伙 i LU D { 字匹配。b G v 以 I 显 > D > X字符集m > D 8 字 Z D字符串。y P D 8 v字Z都G P效D，R X须h置。如果C字段置为全c，则 **fqplu_name** 字段 + C Z 指定需要D伙 i LU 。

mode_name

在配置过L中定义D一组网g X性D { F。b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * <)， I EBCDIC U q R n d。

fqplu_name

伙 i LU D完 { 7 6 LU D { 字。{ F $ 度为 17 v字Z，" 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(? v { F D $ 度最多I为 8 v字Z，R ；含6入U q。）v 1 **plu_alias** 字段h置为全c 1，| E P效。

polarity

会话D相反k s。I能D值P:

AP_POL_EITHER
AP_POL_FIRST_SPEAKER
AP_POL_BIDDER

如果选择K AP_POL_EITHER，ACTIVATE_SESSION 激活I能D发话方会话；否则，激活k s方会话。如果P k s相反会话D I 能，则对Z AP_POL_FIRST_SPEAKER 或 AP_POL_BIDDER，ACTIVATE_SESSION v v在此1 I功。

cnos_permitted

C字段I置为 AP_YES 或 AP_NO 。如果因X定模 = D会话限制4位，而9新会话D激活；I能，" R C字段置为 AP_YES ,那4 L序u < 化隐 = CNOS 处理来u < 化会话限制。1 CNOS 处理发z 1，C动词D执行 + 会暂停。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

secondary_rc

AP_AS_SPECIFIED

AP_AS_NEGOTIATED

session_id
激活会话D 8 字Z m > 符。

如果C动词因N } 错误而未能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_EXCEEDS_MAX_ALLOWED

AP_INVALID_CNOS_PERMITTED
AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS
AP_INVALID_MODE_NAME
AP_INVALID_PLU_NAME

如果C动词，过K C模 = D会话限制，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

Secondary_rc
AP_EXCEEDS_MAX_ALLOWED

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c 停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

如果C动词因其 | 错误而未能执行，则L序返回下P N }:

primary_rc
AP_ACTIVATION_FAIL_NO_RETRY

AP_ACTIVATION_FAIL_RETRY

DEACTIVATE_CONV_GROUP

DEACTIVATE_CONV_GROUP 动词 k s k 指定对话组相一致 D 会话 D M放。！管 C 动词 G Z c Y 作员功能 API D 一？分，｜主要计划 C Z & C L 序 L 序员 ` 写 9 C v 人通信或通信服务器 APPC API D B 务 L 序。 I 在 *v 人通信M户机/服务器通信 L 序 h 计*中 h v D MC_ALLOCATE、ALLOCATE、MC_GET_ATTRIBUTES、GET_ATTRIBUTES 和 RECEIVE_ALLOCATE 动词返回 D 对话组 j 6 符。

VCB a 构

```
typedef struct deactivate_conv_group
{
        unsigned short  opcode;         /* verb operation code           */
        unsigned char   reserv2;        /* reserved                      */
        unsigned char   format;         /* format                        */
        unsigned short  primary_rc;     /* primary return code           */
        unsigned long   secondary_rc;   /* secondary return code         */
        unsigned char   lu_name[8];     /* local LU name                 */
        unsigned char   lu_alias[8];    /* local LU alias                */
        unsigned long   conv_group_id;  /* conversation group identifier */
        unsigned char   type;           /* deactivation type             */
        unsigned char   reserv3[3];     /* reserved                      */
        unsigned long   sense_data;     /* deactivation sense data       */
} DEACTIVATE_CONV_GROUP;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_DEACTIVATE_CONV_GROUP

format

j 6 VCB D q = 。 + C 字段 h 置为 c，以指定 O f P v D VCB D f > 。

lu_name

> X LU k s C Z M放对话组 D LU { 。 C { F G v 8 字 Z D A 型 EBCDIC 字符串。如果 C 字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU 。

lu_alias

> X LU k s C Z M放对话组 D p { 。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 v 1 **lu_name** 字段 h 置为全 c 1，｜E P 效，在 b 种 i v 下。 y P 8 v 字 Z 都 G P 效 D " R X 须 h 置。如果 **lu_name** 和 **lu_alias** = v 字段置为全 c，则 C 动词转发 x k X 制 c 相关 * D LU（缺！ LU）。

conv_group_id

C Z 会话M放 D 对话组 j 6 符。

type M放类型。C 字段 G I M放类型 k 指 > 动词 G 否同 = 或；同 = D j 志位 x 行或运 c D 位掩 k 组 I 。

M放类型:

AP_DEACT_CLEANUP

会话 " 即终止，；H 待从伙 i LU 来 D & 答。

AP_DEACT_NORMAL

在 y P 9 C C 会话 D 对话 a x 后，C 会话终止。

DEACTIVATE_CONV_GROUP

动词运行 i v:

AP_ASYNCHRONOUS_DEACTIVATION
C 动词 " 即返回。

AP_SYNCHRONOUS_DEACTIVATION
C 动词 v 在会话M放后返回。

sense_data
指定在M放D CLEANUP 类型中 9 C D检 b }] 。

返回N数

如果 C 动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_CLEANUP_TYPE

AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS

如果 C 动词因 Z c P 未启动而未能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果 C 动词因 Z c 停止而未能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果 C 动词因系统错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEACTIVATE_SESSION

DEACTIVATE_SESSION 动词k s X定会话，或在y P X定模=中D会话DM放。

VCB a构

```
typedef struct deactivate_session
{
        unsigned short  opcode;         /* verb operation code     */
        unsigned char   reserv2;        /* reserved                */
        unsigned char   format;         /* format                  */
        unsigned short  primary_rc;     /* primary return code     */
        unsigned long   secondary_rc;   /* secondary return code   */
        unsigned char   lu_name[8];     /* local LU name           */
        unsigned char   lu_alias[8];    /* local LU alias          */
        unsigned char   session_id[8];  /* session identifier      */
        unsigned char   plu_alias[8];   /* partner LU alias        */
        unsigned char   mode_name[8];   /* mode name               */
        unsigned char   type;           /* deactivation type       */
        unsigned char   reserv3[3];     /* reserved                */
        unsigned long   sense_data;     /* deactivation sense data */
        unsigned char   fqplu_name[17]; /* fully qualified partner */
                                        /* LU name                 */
        unsigned char   reserv4[20];    /* reserved                */
} DEACTIVATE_SESSION;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEACTIVATE_SESSION

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

lu_name

>X LU k s C Z M放会话D LU D{字。C{F G v 8 字Z D A 型EBCDIC 字符串。如果C字段置为全c，则 **lu_alias** 字段+C Z v定>X LU。

lu_alias

>X LU k s C Z M放会话D p{。b G v以I显>D>X字符集m>D 8 字Z D字符串。 v 1 **lu_name** 字段h置为全c 1，|E P效，在b种i v下。y P 8 v字Z都G P效D " R X须h置。如果 **lu_name** 和 **lu_alias** =v字段置为全c，则C动词转发x k X制c相关*D LU（缺！ LU)。

session_id

C Z M放会话D 8 v字Z j 6符。如果C字段置为全c，则v人通信或通信服务器M放伙i LU 和模=D全？会话。

plu_alias

为>X LU y知D伙i LU D p{。b v{字X须k在配置过L中(" D伙i LU D{字匹配。b G v以I显>D>X字符集m>D 8 字Z D字符串。y P D 8 v字Z都G P效D，R X须h置。如果C字段置为全c，则 **fqplu_name** 字段C Z指定y需要D LU 。

DEACTIVATE_SESSION

mode_name

在配置过L中定义D一组网g X性D{ F。 b G v 8 字Z D字母}字集D A 型EBCDIC 字符串（以一v字母* <）， I EBCDIC U q R n d。

type M放类型。 C字段G I M放类型k指> 动词G否同= 或；同= D j 志位x行或运c D位掩k组I。

M放类型:

AP_DEACT_CLEANUP

会话"即终止， ；H待从伙i LU 来D&答。

AP_DEACT_NORMAL

在y P 9 C C会话D对话a x后， C会话终止。

动词运行i v:

AP_ASYNCHRONOUS_DEACTIVATION

C动词"即返回。

AP_SYNCHRONOUS_DEACTIVATION

C动词v在会话M放后返回。

sense_data

指定在M放D CLEANUP 类型中9 C D检b }]。

fqplu_name

伙i LU D完{ 7 6 LU {。{ F $度为17 v字Z， "以EBCDIC U q R n d。 | I = v A 型EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v{ F D $度最多I为8 v字Z， R；含6入U q。）v 1 **plu_alias** 字段h置为全c 1， | E P效。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

注意如果 **session_id** ；能k k任何1 0会话相匹配，则假h b G因为会话已M放。在b种i v下动词I功a x。

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_MODE_NAME

AP_INVALID_PLU_NAME
AP_INVALID_CLEANUP_TYPE
AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

PATH_SWITCH

PATH_SWITCH 动词 k s v 人通信或通信服务器 C Z 在支 V _ 性能 7 I 选择（HPR）D ，S 中 P 换 7 I。 如果；能 R = | 好 D 7 6， ，S + ；做 D d D 继续。

VCB a 构

```
typedef struct path_switch
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   rtp_connection_name[8];
                                        /* RTP connection name   */
} PATH_SWITCH;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_PATH_SWITCH

format

j 6 VCB D q =。+ C 字段 h 置为 c，以指定 O f P v D VCB D f >。

rtp_connection_name

j 6 = 7 6 * 关 D RTP ，S。b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。y P D 8 v 字 Z 都 G P 效 D， R X 须 h 置。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_RTP_CONNECTION

如果 C 动词因状，错误而未能执行，则 L 序返回下 P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_PATH_SWITCH_IN_PROGRESS

如果 C 动词因 7 6 P 换 T 图 ' \ ，则 L 序返回下 P N }:

primary_rc
AP_UNSUCCESSFUL

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

第6章 i 询动词

> B h v K C Z i 询 P 关 Z c 配置和状，D 信息 D 动词。

SNA API M户机 O v 支 V 某些 N } 。

k i 4 > B 中 D *记 B* > 图 j， 以获 C 详细信息。

获 C 详细信息。

© Copyright IBM Corp. 1989, 1997, 1998

format

j 6 VCB D q = 。 + C字段 h 置为 c , 以指定O f P v D VCB D f > 。

buf_ptr

指向C Z写入P m信息D缓 e x D指 k 。 & C L序 I + }] m加 = VCB D末尾（X须 + **buf_ptr** h 置为 NULL)。

buf_size

y a 供D缓 e x D大小。返回D }] + ; , 过 b v 大小。

num_entries

要返回D项D最大 } ? 。项D } ? ; 要 , 过 b v 值。 c 值m > 无限制。

list_options

| 指 > 在P m信息中&返回 2 4 : 指定D **adj_nncp_name**（见下P N } ）m > w引值, C w引值C Z 指定要返回D 5 际信息D起 < c 。

AP_FIRST_IN_LIST

忽T w引值, 返回 P m从 P m D Z 一项 * < 。

AP_LIST_FROM_NEXT

返回 P m从 P m中 4 a 供D w引值 y 指定D那项D下一项 * < 。

AP_LIST_INCLUSIVE

返回 P m从w引值 y 指定D那项 * < 。

adj_nncp_name

Z | 网 g Z c 17 v 字 Z $ D全限定 { F 。 | G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串组 I D , " C EBCDIC U q R n d 。(? v { F D $ 度最多 I 为 8 v 字 Z , " ; 含 6 入 U q 。）如果 **list_options** h 置为 AP_FIRST_IN_LIST, 则忽T C字段。

返回N数

如果动词 I 功执行, 则L 序返回下 P N } :

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大 Z **buf_size**。

num_entries

5 际返回D项D } ? 。

total_num_entries

已 - 返回D项D总 } 。 C值 I 以大 Z **num_entries**。

adj_nncp_data.overlay_size

C项中D字 Z } , 即 = 下一 v 返回项（如果 P D话）D偏移 ? 。

QUERY_ADJACENT_NN

adj_nncp_data.adj_nncp_name
Z | 网 g Z c 17 v 字 Z $ D 全限定 { F 。 | G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串组 I D, " C EBCDIC U q R n d 。(? v { F D $ 度最多 I 为 8 v 字 Z, " ; 含 6 入 U q 。)

adj_nncp_data.cp_cp_sess_status
CP-CP 会话 D 状 , 。 | h 置为下 P 之一:

AP-ACTIVE
AP_CONWINNER_ACTIVE
AP_CONLOSER_ACTIVE
AP_INACTIVE

adj_nncp_data.out_of_seq_tdus
S U = 来自 C Z c D 无序 TDU D } ? 。

adj_nncp_data.last_frsn_sent
O 一 v 发 M 至 C Z c D w 减 Y 序号。

adj_nncp_data.last_frsn_rcvd
O 一 v S U = 自 C Z c D w 减 Y 序号。

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_ADJ_NNCP_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_ADJACENT_NODE

QUERY_ADJACENT_NODE 返回P关在 DEFINE_ADJACENT_NODE O配置D Z | Z c D信息。

信息在一E排序D P m中返回。 P m中D ? v项I一v | 含Z | CP 信息D ADJACENT_NODE_DATA 2G，后z k C Z | CP P关D ? v LU D ADJACENT_NODE_LU_DATA 2G组I。

先4 **cp_name** 排P项，然后4 **fqlu_name** 排序。W先4 { F $度排序，然后对相同$度D { F x行 ASCII 词d `辑排序（y] j准 MIB 定序)。

如果选中K AP_LIST_FROM_NEXT，则P m 4定义D次序（无[指定D项G否存在）从下一项* <。

VCB a构

```
typedef struct query_adjacent_node
{
        unsigned short  opcode;            /* Verb operation code        */
        unsigned char   reserv2;           /* reserved                   */
        unsigned char   format;            /* format                     */
        unsigned short  primary_rc;        /* Primary return code        */
        unsigned long   secondary_rc;      /* Secondary return code      */
        unsigned char   *buf_ptr;          /* pointer to buffer          */
        unsigned long   buf_size;          /* buffer size                */
        unsigned long   total_buf_size;    /* total buffer size required */
        unsigned short  num_entries;       /* number of entries          */
        unsigned short  total_num_entries; /* total number of entries    */
        unsigned char   list_options;      /* listing options            */
        unsigned char   reserv3;           /* reserved                   */
        unsigned char   cp_name[17];       /* CP name of adjacent node   */
} QUERY_ADJACENT_NODE;

typedef struct adjacent_node_data
{
        unsigned short  overlay_size;      /* size of this entry         */
        unsigned short  sub_overlay_size;  /* size of this stub entry    */
        unsigned char   cp_name[17];       /* CP name                    */
        DESCRIPTION     description;       /* resource description       */
        unsigned char   reserv3[19];       /* reserved                   */
        unsigned short  num_of_lus;        /* number of LUs              */
} ADJACENT_NODE_DATA;

typedef struct cn_det_data
{
        unsigned short  num_act_ports;     /* number of active ports     */
        unsigned char   reserva[20];       /* reserved                   */
} CN_DET_DATA;

typedef struct cn_def_data
{
        unsigned char   description[RD_LEN];
                                           /* resource description       */
        unsigned char   num_ports;         /* number of ports on CN      */
        unsigned char   reserv1[16];       /* reserved                   */
        TG_DEFINED_CHARS tg_chars;         /* TG characteristics         */
} CN_DEF_DATA;
```

QUERY_ADJACENT_NODE

```
typedef struct adjacent_node_lu_data
{
        unsigned short  overlay_size;      /* effective capacity         */
        unsigned char   reserve2[2];       /* reserved                   */
        ADJACENT_NODE_LU adj_lu_def_data;  /* Adjacent LU defined data   */
} ADJACENT_NODE_LU_DATA;

typedef struct adjacent_node_lu
{
        unsigned char   wildcard_lu;       /* Is this LU a wildcard?     */
        unsigned char   fqlu_name[17];     /* Fully-Qualified LU name    */
        unsigned char   reserve1[6];       /* reserved                   */
        ADJACENT_NODE_LU adj_lu_def_data;  /* Adjacent LU defined data   */
} ADJACENT_NODE_LU;
```

提供N数

&CL序a供下PN}:

opcode

AP_QUERY_ADJACENT_NODE

format

j 6 VCB Dq =。+C字段h置为c，以指定Of Pv D VCB Df >。

buf_ptr

指向CZ写入Pm信息D缓e x D指k。&CL序I + }]m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D}] + ; ,过b v大小。

num_entries

要返回D项D最大}?。项D}?;要,过b v值。 c值m >无限制。

list_options

mw在Pm信息中&C返回24。指定D **cp_name**（见下PN}）m > w引值， Cw引值CZ指定要返回D5际信息D起< c。

AP_FIRST_IN_LIST

忽Tw引值，返回Pm从L序维护D目<中DZ一v Z | Z c * <。

AP_LIST_FROM_NEXT

返回Pm从Pm中4 a供Dw引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回Pm从w引值y指定D那项* <。

cp_name

Z | Z cD全限定{ F。C{ FGI = v C EBCDIC c串* D A型EBCDIC字符串构I, " C EBCDIC Uq R n d。(? v{ FD $度最多I为8 v字Z, " ;含6入Uq。)

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc

AP_OK

buf_size

在缓 e x 中返回 D 信息 $ 度。

total_buf_size

返回值 m w 返回 y P ; k s D P m 信息 y 需要 D 缓 e x D 大小。C 值 I 以大 Z **buf_size**。

num_entries

5 际返回 D 项 D } ?。

total_num_entries

已 - 返回 D 项 D 总 }。C 值 I 以大 Z **num_entries**。

adjacent_node_data.overlay_size

C 项中 D 字 Z }（ | 括任何 ADJACENT_NODE_LU_DATA a 构），即 = 下一 v 返回项（如果 P D 话）D 偏移 ?。

adjacent_node_data.sub_overlay_size

C 项中 Z c ? 分 D 字 Z }, ; | 括任何 ADJACENT_NODE_LU_DATA a 构; b G = C 项 D Z 一 v ADJACENT_NODE_LU_DATA 字段 D 偏移 ?。

adjacent_node_data.cp_name

Z | Z c D 全限定 { F。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I, " C EBCDIC U q R n d。(? v { F D $ 度最多 I 为 8 v 字 Z, " ; 含 6 入 U q。)

cn_data.det_data.num_act_ports

a 供，S 网 g O 活动端 Z } D 动，值。

adjacent_node_data.description

资源 5 w（如在 DEFINE_ADJACENT_NODE 中 5 w D)。C 字段 D $ 度 & 为 4 字 Z D 6 }, R ; 为 c。

adjacent_node_data.num_of_lus

为 C Z | Z c 定义 D LU D } ?。后 z ? v LU D ADJACENT_NODE_LU_DATA a 构。

adjacent_node_lu_data.overlay_size

C 项中 D 字 Z }, 即 = 下一 v 返回项（如果 P D 话）D 偏移 ?。

adjacent_node_lu_data.adj_lu_def_data.wildcard_lu

m w LU { F G 否定义为通配符。

adjacent_node_lu_data.adj_lu_def_data.fqlu_name

Z | Z c D 全限定 { F。{ F $ 度为 17 v 字 Z, " 以 EBCDIC U q R n d。(? v { F D $ 度最多 I 为 8 v 字 Z, " ; 含 6 入 U q。) C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I, " C EBCDIC U q R n d。(? v { F D $ 度最多 I 为 8 v 字 Z, " ; 含 6 入 U q。)

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_CP_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_CN

QUERY_CN 返回 P 关 Z ｜ ，S 网 g D信息。 C信息 I 『确定 }] 』（执行期间动，U 集 D }] ）和『定义 }] 』（DEFINE_CN O D & C L 序 y a 供 D }] ）构 I 。

b 些信息以 q = 化 P m返回。如需 P 关某 v X定 CN D信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **fqcn_name** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。 k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **fqcn_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。

如果选中 K AP_LIST_FROM_NEXT，则 P m 4 定义 D次序（无 [指定 D项 G 否存在）从下一项 * < 。

VCB a 构

```
typedef struct query_cn
{
        unsigned short  opcode;            /* Verb operation code        */
        unsigned char   attributes;        /* verb attributes            */
        unsigned char   format;            /* format                     */
        unsigned short  primary_rc;        /* Primary return code        */
        unsigned long   secondary_rc;      /* Secondary return code      */
        unsigned char   *buf_ptr;          /* pointer to buffer          */
        unsigned long   buf_size;          /* buffer size                */
        unsigned long   total_buf_size;    /* total buffer size required */
        unsigned short  num_entries;       /* number of entries          */
        unsigned short  total_num_entries; /* total number of entries    */
        unsigned char   list_options;      /* listing options            */
        unsigned char   reserv3;           /* reserved                   */
        unsigned char   fqcn_name[17];     /* Name of connection network */
} QUERY_CN;

typedef struct cn_data
{
        unsigned short  overlay_size;      /* size of this entry         */
        unsigned char   fqcn_name[17];     /* Name of connection network */
        unsigned char   reserv1;           /* reserved                   */
        CN_DET_DATA     det_data;          /* Determined data            */
        CN_DEF_DATA     def_data;          /* Defined data               */
} CN_DATA;

typedef struct cn_det_data
{
        unsigned short  num_act_ports;     /* number of active ports     */
        unsigned char   reserva[20];       /* reserved                   */
} CN_DET_DATA;

typedef struct cn_def_data
{
        unsigned char   description[RD_LEN];
                                           /* resource description       */
        unsigned char   num_ports;         /* number of ports on CN      */
        unsigned char   reserv1[16];       /* reserved                   */
        TG_DEFINED_CHARS tg_chars;         /* TG characteristics         */
} CN_DEF_DATA;

typedef struct tg_defined_chars
{
        unsigned char   effect_cap;        /* effective capacity         */
```

```
        unsigned char   reserve1[5];       /* reserved                    */
        unsigned char   connect_cost;      /* connection cost             */
        unsigned char   byte_cost;         /* byte cost                   */
        unsigned char   reserve2;          /* reserved                    */
        unsigned char   security;          /* security                    */
        unsigned char   prop_delay;        /* propagation delay           */
        unsigned char   modem_class;       /* modem class                 */
        unsigned char   user_def_parm_1;   /* user-defined parameter 1    */
        unsigned char   user_def_parm_2;   /* user-defined parameter 2    */
        unsigned char   user_def_parm_3;   /* user-defined parameter 3    */
} TG_DEFINED_CHARS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_CN

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I S性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m >无限制。

list_options

|指>在P m信息中&返回2 4: 指定D **fqcn_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

fqcn_name

17 v字Z D全限定，S网g { F。C { F G I = v C EBCDIC c串* D A型EBCDIC 字符串构I， " C EBCDIC U q R n d。(? v { F D $度最多I为8 v字Z， " ；含6入U q。)如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓 e x中返回D信息$度。

total_buf_size
返回值m w返回 y P ; k s D P m信息y需要D缓 e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**。

cn_data.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

cn_data.fqcn_name
17 v字Z D全限定，S网g { F。C { F G I = v C EBCDIC c串* D A型EBCDIC 字符串构I， " C EBCDIC U q R n d。（? v { F D $度最多I为8 v字Z， " ；含6人U q。）

cn_data.det_data.num_act_ports
a供，S网g O活动端Z } D动，值。

cn_data.def_data.description
资源5 w（如在 DEFINE_CN 中5 w D）。| G一v以> X I显>字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

cn_data.def_data.num_ports
，S网g O D端Z }。

cn_data.def_data.tg_chars.effect_cap
P效能&D 5际%元。C值 ` k为一v 1 字Z D ! c }， I公= 0.1mmm * 2 eeeee m >，其中字Z D位m >为 eeeeemmm。P效能&D ? v %元H Z 300 位 / k。

cn_data.def_data.tg_chars.connect_cost（每，S时d成本）
P效值G 0--255 之间D { }值，其中 0 G最M ?，S 1间I >，而 255 G最 _ ?，S 1间I >。

cn_data.def_data.tg_chars.byte_cost
?字Z D I >。P效值G 0--255 之间D { }值，其中 0 G最M ?字Z I >，而 255 G最 _ ?字Z I >。

cn_data.def_data.tg_chars.security
2全性值如下f P m y >。

AP_SEC_NONSECURE
无2全性。

AP_SEC_PUBLIC_SWITCHED_NETWORK
在C ， S网 g O发M D }] + 在公C ；换网 g O w动。

AP_SEC_UNDERGROUND_CABLE
}] 在 2 全X下 g 缆中发M。

AP_SEC_SECURE_CONDUIT
线 7 G ；加 # 护D 2 全护线管。

AP_SEC_GUARDED_CONDUIT
护线管防止物理 e 击。

AP_SEC_ENCRYPTED
在线 7 O加 \ 。

AP_SEC_GUARDED_RADIATION
线 7 防止物理和辐 d e 击。

cn_data.def_data.tg_chars.prop_delay
传%延Y， m > 一 v 信号在 4 7 中传M D 1 间， %位为微 k 。 C 值 ` k 为一 v 1 字 Z D ！ c } ， I 公 = 0.1mmm * 2 eeeee m > ， 其中字 Z D 位 m > 为 eeeeemmm。缺！值如下 y > 。

AP_PROP_DELAY_MINIMUM
无传%延Y。

AP_PROP_DELAY_LAN
小Z 480 微 k 延Y。

AP_PROP_DELAY_TELEPHONE
在 480 和 49 512 微 k 间延Y。

AP_PROP_DELAY_PKT_SWITCHED_NET
在 49 512 和 245 760 微 k 间延Y。

AP_PROP_DELAY_SATELLITE
大 Z 245 760 微 k 延Y。

AP_PROP_DELAY_MAXIMUM
最大传%延Y。

cn_data.def_data.tg_chars.modem_class
t 。 C 字段& < 终 h 置为 c 。

cn_data.def_data.tg_chars.user_def_parm_1
在 0--255 间D C 户定义N } 。

cn_data.def_data.tg_chars.user_def_parm_2
在 0--255 间D C 户定义N } 。

cn_data.def_data.tg_chars.user_def_parm_3
在 0--255 间D C 户定义N } 。

如果因N } 错误而 9 动词 ; 能执行， 则L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_CN_NAME

AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_CN_PORT

QUERY_CN_PORT 返回 P 关在 Z | ，S 网 g O定义 D端 Z D信息。 b 些信息以 q = 化 P m返回。如需 P 关某 v X定端 Z D信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **port_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。注意: **fqcn_name** 字段 X 须 < 终 h 置为 P 效，S 网 g D { F。

k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6。

VCB a 构

```
typedef struct query_cn_port
{
        unsigned short  opcode;            /* Verb operation code        */
        unsigned char   reserv2;           /* reserved                   */
        unsigned char   format;            /* format                     */
        unsigned short  primary_rc;        /* Primary return code        */
        unsigned long   secondary_rc;      /* Secondary return code      */
        unsigned char   *buf_ptr;          /* pointer to buffer          */
        unsigned long   buf_size;          /* buffer size                */
        unsigned long   total_buf_size;    /* total buffer size required */
        unsigned short  num_entries;       /* number of entries          */
        unsigned short  total_num_entries; /* total number of entries    */
        unsigned char   list_options;      /* listing options            */
        unsigned char   reserv3;           /* reserved                   */
        unsigned char   fqcn_name[17];     /* Name of connection network */
        unsigned char   port_name[8];      /* port name                  */
} QUERY_CN_PORT;

typedef struct cn_port_data
{
        unsigned short  overlay_size;       /* size of this entry         */
        unsigned char   fqcn_name[17];      /* Name of connection network */
        unsigned char   port_name[8];       /* name of port               */
        unsigned char   tg_num;             /* transmission group number  */
        unsigned char   reserva[20];        /* reserved                   */
} CN_PORT_DATA;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_CN_PORT

format

j 6 VCB D q =。+ C字段 h 置为 c，以指定 O f P v D VCB D f >。

buf_ptr

指向 C Z 写入 P m信息 D缓 e x D指 k。& C L 序 I + }] m加 = VCB D末尾（X 须 + **buf_ptr** h 置为 NULL）。

buf_size

y a 供 D缓 e x D大小。返回 D }] + ; ，过 b v 大小。

num_entries

要返回 D 项 D最大 } ?。项 D } ? ；要，过 b v 值。 c 值 m > 无限制。

list_options

| 指 > 在 P m 信息中 & 返回 2 4 : 指定 D **fqcn_name** 和 **port_name** D 组合（见下 P N } ）m > w 引值， C w 引值 C Z 指定要返回 D 5 际信息 D 起 < c 。

AP_FIRST_IN_LIST

忽 T w 引值，返回 P m 从 P m D Z 一项 * < 。

AP_LIST_FROM_NEXT

返回 P m 从 P m 中 4 a 供 D w 引值 y 指定 D 那项 D 下一项 * < 。

AP_LIST_INCLUSIVE

返回 P m 从 w 引值 y 指定 D 那项 * < 。

fqcn_name

17 v 字 Z D 全限定，S 网 g { F 。 C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I , " C EBCDIC U q R n d 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , " ；含 6 入 U q 。）X 须 < 终 h 置 C 字段。

port_name

以 > X I 显 > 字符集 m > D 8 字 Z 字符串。 y P 8 v 字 Z 都 G P 效 D , " R X 须 h 置。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽 T C 字段。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N } :

primary_rc

AP_OK

buf_size

在缓 e x 中返回 D 信息 $ 度。

total_buf_size

返回值 m w 返回 y P ； k s D P m 信息 y 需要 D 缓 e x D 大小。 C 值 I 以大 Z **buf_size**。

num_entries

5 际返回 D 项 D } ? 。

total_num_entries

已 - 返回 D 项 D 总 } 。 C 值 I 以大 Z **num_entries**。

cn_port_data.overlay_size

C 项中 D 字 Z } , 即 = 下一 v 返回项（如果 P D 话）D 偏移 ? 。

cn_port_data.fqcn_name

17 v 字 Z D 全限定，S 网 g { F 。 C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I , " C EBCDIC U q R n d 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , " ；含 6 入 U q 。）

cn_port_data.port_name

以 8 字 Z > X I 显 > 字符集 m > D 端 Z { F 。 y P D 8 v 字 Z G P 效 D 。

cn_port_data.tg_num

指定端 Z D 传 d 组号。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N } :

QUERY_CN_PORT

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_CN_NAME

AP_INVALID_PORT_NAME
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_CONVERSATION

QUERY_CN_PORT 返回 P 关在指定 LU O运行D对话D信息。如需 P 关某 v X定对话 D信息，或要获C几 v 『段』中D P m信息， & h置 **conv_id** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。注意: X须 < 终 h 置 **lu_alias** 字段。如果 lu_name 非 c，则 + E 先 9 C |，而；G lu_alias。

k N阅 Z 12页D『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **conv_id** 排序。如果选中K AP_LIST_FROM_NEXT, 则返回D P m 4 w 引（无 [指定D项G否存在）从下一项 * < 。

VCB a 构

```
typedef struct query_conversation
{
        unsigned short  opcode;            /* Verb operation code        */
        unsigned char   reserv2;           /* reserved                   */
        unsigned char   format;            /* format                     */
        unsigned short  primary_rc;        /* Primary return code        */
        unsigned long   secondary_rc;      /* Secondary return code      */
        unsigned char   *buf_ptr;          /* pointer to buffer          */
        unsigned long   buf_size;          /* buffer size                */
        unsigned long   total_buf_size;    /* total buffer size required */
        unsigned short  num_entries;       /* number of entries          */
        unsigned short  total_num_entries; /* total number of entries    */
        unsigned char   list_options;      /* listing options            */
        unsigned char   reserv3;           /* reserved                   */
        unsigned char   lu_name[8];        /* local LU name              */
        unsigned char   lu_alias[8];       /* local LU alias             */
        unsigned long   conv_id;           /* conversation identifier    */
        unsigned char   session_id[8];     /* session identifier         */
        unsigned char   reserv4[12];       /* reserved                   */
} QUERY_CONVERSATION;

typedef struct conv_summary
{
        unsigned short  overlay_size;      /* size of this entry         */
        unsigned long   conv_id;           /* conversation identifier    */
        unsigned char   local_tp_name[64]; /* Name of local TP           */
        unsigned char   partner_tp_name[64];
                                           /* Name of partner TP         */
        unsigned char   tp_id[8];          /* TP identifier              */
        unsigned char   sess_id[8];        /* session identifier         */
        unsigned long   conv_start_time;   /* time conversation was      */
                                           /* started                    */
        unsigned long   bytes_sent;        /* bytes sent so far          */
        unsigned long   bytes_received;    /* bytes received so far      */
        unsigned char   conv_state;        /* conversation state         */
        unsigned char   duplex_type;       /* conversation duplex type   */
} CONV_SUMMARY;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_CONVERSATION

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

QUERY_CONVERSATION

buf_ptr

指向C Z写入P m信息D缓 e x D指 k。 &C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓 e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4。指定D **index**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值 y 指定D那项 * <。

lu_name

> X LU D { F。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（；以 } 字 * <），I EBCDIC U q R n d。

lu_alias

p {， > X TP 利C | I以知@ > X LU。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须h置。

conv_id

对话 ID。

session_id

如果 | 为全二x制c，则C字段；能C Z过K返回对话。如果 | 非c，则v返回那些其会话 ID k y a供D值相匹配D对话。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓 e x中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 C值I以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总 }。 C值I以大Z **num_entries**。

conv_summary.overlay_size
C项中D字Z }, 即=下一v返回项（如果P D话）D偏移？。

conv_summary.conv_id
对话 ID。

C N } D值I w C B务Y作中D ALLOCATE 动词或；w C D B务L序中D RECEIVE_ALLOCATE 返回。

conv_summary.local_tp_name
> X B务L序D { F。

conv_summary.partner_tp_name
伙i B务L序D { F。 | v J C Z > X启动D对话。对Z远L启动D对话, | 为U。

conv_summary.tp_id
指定x B务L序D B务L序j 6符。C j 6符；G I API P S e指定, M G I NOF B务L序管理器指定。

conv_summary.sess_id
分配x C对话D会话j 6符。

conv_summary.conv_start_time
从启动Z c =启动对话y - 过D 1间, 以Y分之一k为%位。

conv_summary.bytes_sent
=目0为止在C对话O y发M D字Z }。

conv_summary.bytes_received
=目0为止在C对话O y S U D字Z }。

conv_summary_conv_state
conv_id y j 6 D对话D 1 0状，。对Z k +工对话, | G下P值之一:

AP_RESET_STATE

AP_SEND_STATE
AP_RECEIVE_STATE
AP_CONFIRM_STATE
AP_CONFIRM_SEND_STATE
AP_CONFIRM_DEALL_STATE
AP_PEND_POST_STAT
AP_PEND_DEALL_STATE
AP_END_CONV_STATE
AP_SEND_PENDING_STATE
AP_POST_ON_RECEIPT_STATE

对Z全+工对话, | G下P值之一:

AP_RESET_STATE
AP_SEND_RECEIVE_STATE
AP_SEND_ONLY_STATE
AP_RECEIVE_ONLY_STATE

conv_summary.duplex_type
指定C对话G k +工还G全+工。

AP_HALF_DUPLEX
AP_FULL_DUPLEX

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_BAD_CONV_ID

AP_INVALID_LU_ALIAS
AP_INVALID_LU_NAME

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_COS

QUERY_COS 返回 P 关 X 定服务级 D 7 I 计 c 信息。 b 些信息以 q = 化 P m返回。如需 P 关某 v X定 COS D信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **cos_name** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C 字段。 k N 阅 Z 12 页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。 C P m y] **cos_name** 排序。W 先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] IBM D 6611 APPN MIB 定序）。如果选中 K AP_LIST_FROM_NEXT，返回 P m 4 定义 D 次序（无 [指定 D 项 G 否存在）从下一项 * < 。

VCB a 构

```
typedef struct query_cos
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   *buf_ptr;          /* pointer to buffer             */
        unsigned long   buf_size;          /* buffer size                   */
        unsigned long   total_buf_size;    /* total buffer size required    */
        unsigned short  num_entries;       /* number of entries             */
        unsigned short  total_num_entries; /* total number of entries       */
        unsigned char   list_options;      /* listing options               */
        unsigned char   reserv3;           /* reserved                      */
        unsigned char   cos_name[8];       /* COS name                      */
} QUERY_COS;

typedef struct cos_data
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   cos_name[8];       /* COS name                      */
        unsigned char   description[RD_LEN];
                                           /* resource description          */
        unsigned char   transmission_priority;
                                           /* transmission priority         */
        unsigned char   reserv1;           /* reserved                      */
        unsigned short  num_of_node_rows;  /* number of node rows           */
        unsigned short  num_of_tg_rows;    /* number of TG rows             */
        unsigned long   trees;             /* number of tree caches for COS */
        unsigned long   calcs;             /* number of route calculations  */
                                           /* for this COS                  */
        unsigned long   rejs;              /* number of route rejects       */
                                           /* for COS                       */
        unsigned char   reserva[20];       /* reserved                      */
} COS_DATA;
```

提供 N 数

& C L 序 a 供下 P N }：

opcode

AP_QUERY_COS

format

j 6 VCB D q = 。 + C 字段 h 置为 c，以指定 O f P v D VCB D f > 。

buf_ptr

指向C Z写入P m信息D缓 e x D指 k。 &C L序 I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓 e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过b v值。 c值m > 无限制。

list_options

|指 > 在P m信息中&返回 2 4: 指定D **cos_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起 < c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中 4 a供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值 y 指定D那项 * <。

cos_name

服务级{ F。 b G一v 8 字Z字母 }字D A 型 EBCDIC 字符串（以一v字母 * 头）， " 以 EBCDIC U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词 I 功执行，则L 序返回下P N }:

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大Z **buf_size**。

num_entries

5 际返回D项D } ?。

total_num_entries

已 - 返回D项D总 }。 C值 I 以大Z **num_entries**。

cos_data.overlay_size

C项中D字Z }，即 = 下一v返回项（如果P D话）D偏移 ?。

cos_data.cos_name

服务级{ F。 b G一v 8 字Z字母 }字D A 型 EBCDIC 字符串（以一v字母 * 头）， " 以 EBCDIC U q R n d。

cos_data.description

资源 5 w（如在 DEFINE_COS 中 5 w D）。 | G一v以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

cos_data.transmission_priority
传 d E 先级。 | h 置为下 P 值之一:

AP_LOW
AP_MEDIUM
AP_HIGH
AP_NETWORK

cos_data.num_of_node_rows
C COS D Z c 行 } 。

cos_data.num_of_tg_rows
C COS D TG 行 } 。

cos_data.trees
自 O 次 u < 化以来，为 C COS （ " D w 型 7 I _ Y 缓存 D } 目。

cos_data.calcs
指定 C 服务级 D 会话激活 k s （因此 M G 7 I 计 c ） D } ? 。

cos_data.rejs
I Z 从 C Z c - 过网 g = | { D 目 D X 之间； P I S \ D （ 9 C 指定服务级 D ） 7 I 而 ' \ D 会话激活 k s D 次 } 。 v 1 一 u 7 I 完全 I 能够 a 供指定服务级 D 活动 TG 和 Z c 组 I 1， | E G I S \ D 。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_COS_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DEFAULT_PU

QUERY_DEFAULT_PU 9 C户能 i 询C DEFINE_DEFAULT_PU 动词 y 定义D缺！PU。

VCB a 构

```
typedef struct query_default_pu
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        unsigned char   def_pu_name[8]; /* default PU name       */
        unsigned char   description[RD_LEN];
                                        /* resource description  */
        unsigned char   def_pu_sess[8]; /* PU name of active     */
                                        /* default session       */
        unsigned char   reserv3[16];    /* reserved              */
} QUERY_DEFAULT_PU;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_DEFAULT_PU

format

j 6 VCB D q =。 + C字段 h 置为 c，以指定O f P v D VCB D f >。

返回N数

如果动词 I 功执行，则L 序返回下 P N }:

primary_rc

AP_OK

def_pu_name

最 | 在 DEFINE_DEFAULT_PU 动词中指定D PU D { F。 b G v 8 字Z D 字母 } 字集D A 型 EBCDIC 字符串（以字母 * < ），I EBCDIC U q R n d。如果；P 发 v DEFINE_DEFAULT_PU 动词，则C字段 + h 置为全 c 。

description

资源 5 w（如在 DEFINE_DEFAULT_PU 中 5 w D ）。 | G 一 v 以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

def_pu_sess

k 1 0 活动D缺！ PU 会话相关D PU { F。如果已定义K缺！ PU, + k 其关 * D会话G；活动D，则C字段 k **def_pu_name** 字段；同。在 b 种 i v 下，v 人通信或通信服务器继续 9 C k 以 0 D缺！ PU 关 * D会话，直 = k y 定义D缺！ PU 关 * D会话 d 为活动。如果；P 活动D PU 会话，则C字段 + h 置为全 c 。

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DEFAULTS

QUERY_DEFAULTS 9 C户能 i 询C DEFINE_DEFAULTS 动词 y 定义D缺！值。

VCB a 构

```
typedef struct query_defaults
{
        unsigned short  opcode;         /* verb operation code   */
        unsigned char   reserv2;        /* reserved              */
        unsigned char   format;         /* format                */
        unsigned short  primary_rc;     /* primary return code   */
        unsigned long   secondary_rc;   /* secondary return code */
        DEFAULT_CHARS   default_chars;  /* default information   */
} QUERY_DEFAULTS;

typedef struct default_chars
{
        unsigned char   description[RD_LEN];
                                        /* resource description  */
        unsigned char   mode_name[8];   /* default mode name     */
        unsigned char   implicit_plu_forbidden;
                                        /* disallow implicit     */
                                        /* PLUs ?                */
        unsigned char   specific_security_codes;
                                        /* generic security      */
                                        /* sense codes           */
        unsigned char   limited_timeout;/* timeout for limited   */
                                        /* sessions              */
        unsigned char   reserv[244];    /* reserved              */
} DEFAULT_CHARS;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_DEFAULTS

format

j 6 VCB D q =。 + C字段 h 置为 c，以指定O f P v D VCB D f >。

返回N数

如果动词 I 功执行，则L 序返回下 P N }:

primary_rc

AP_OK

default_chars.description

资源 5 w（如在 DEFINE_DEFAULTS 中 5 w D）。 | G一 v 以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

default_chars.mode_name

最 | 在 DEFINE_DEFAULTS 动词中指定D方 = D { F。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。如果； P 发 v DEFINE_DEFAULTS 动词，则C字段 + h 置为全 c。

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DIRECTORY_ENTRY

QUERY_DIRECTORY_LU 从目 < }] b 中返回 LU P m。 b 些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关某 v X定 LU D信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **resource_name** 和 **resource_type** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。 k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

1 > X Z c G网 g Z c 1，信息返回如下:

```
1st Network Node
                        1st LU located at Nework Node
                        2nd LU locate at Network Node
                        ...
                        nth LU located at Network Node
            1st End Node served by this Network Node
                        1st LU located at End Node(1)
                        2nd LU located at End Node(1)
                        ...
                        nth LU located at End Node(1)
             ...
             nth End Node served by this Network Node
                        1st LU located at End Node(n)
                        2nd LU located at End Node(n)
                        ...
2nd Network Node
            ...etc..
```

1 L序以端 Z c Y作 1，资源 P m中返回 D Z 一 v 项 G EN CP。（对 Z 端 Z c D网 g Z c 服务器， ；返回任何项。）

C 返回 D 目 < 项 P m I 以 y] 8 代 { F （和类型）过 K。在 b 种 i v 下， & h 置 **parent_name** 和 **parent_type** 字段（否则， b = v 字段都 & h 置为全 c ）。W先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] IBM D 6611 APPN MIB 定序）。如果选中 K AP_LIST_FROM_NEXT，则返回 D P m 4 定义 D 次序（无 [指定 D 项 G 否存在）从下一项 * < 。

VCB a 构

格式 1

```
typedef struct query_directory_entry{
        unsigned short  opcode;             /* verb operation code         */
        unsigned char   reserv2;            /* reserved                    */
        unsigned char   format;             /* format                      */
        unsigned short  primary_rc;         /* primary return code         */
        unsigned long   secondary_rc;       /* secondary return code       */
        unsigned char   *buf_ptr;           /* pointer to buffer           */
        unsigned long   buf_size;           /* buffer size                 */
        unsigned long   total_buf_size;     /* total buffer size required  */
        unsigned short  num_entries;        /* number of entries           */
        unsigned short  total_num_entries;  /* total number of entries     */
        unsigned char   list_options;       /* listing options             */
```

```
        unsigned char   reserv3;            /* reserved                    */
        unsigned char   resource_name[17];  /* network qualified res name  */
        unsigned char   reserv4;            /* reserved                    */
        unsigned short  resource_type;      /* Resource type               */
        unsigned char   parent_name[17];    /* parent name filter          */
        unsigned char   reserv5;            /* reserved                    */
        unsigned short  parent_type;        /* parent type                 */
        unsigned char   reserv6[24];        /* reserved                    */
 } QUERY_DIRECTORY_ENTRY;

typedef struct directory_entry_summary
{
        unsigned short  overlay_size;       /* size of this entry           */
        unsigned char   resource_name[17];  /* network qualified res name   */
        unsigned char   reserve1;           /* reserved                     */
        unsigned short  resource_type;      /* Resource type                */
        unsigned char   description[RD_LEN]; /* resource description        */
        unsigned char   real_owning_cp_type; /* real owning CP type         */
        unsigned char   real_owning_cp_name[17];
                                            /* real owning CP name          */
} DIRECTORY_ENTRY_SUMMARY;

typedef struct directory_entry_detail
{
        unsigned short  overlay_size;       /* size of this entry          */
        unsigned char   resource_name[17];  /* network qualified res name  */
        unsigned char   reserv1a;           /* reserved                    */
        unsigned short  resource type;      /* Resource type               */
        unsigned char   description[RD_LEN]; /* resource description       */
        unsigned char   parent_name[17];    /* network qualified           */
                                            /* parent name                 */
        unsigned char   reserv1b;           /* reserved                    */
        unsigned short  parent_type;        /* parent resource type        */
        unsigned char   entry_type;         /* Type of the directory entry */
        unsigned char   location;           /* Resource location           */
        unsigned char   real_owning_cp_type; /* real owning CP type        */
        unsigned char   real_owning_cp_name[17];                           */
                                            /* real owning CP name         */
        unsigned char   reserva;            /* reserved                    */
} DIRECTORY_LU_DETAIL;
```

VCB a 构

格式 0（后备 6 别）

```
typedef struct query_directory_entry{
        unsigned short  opcode;             /* verb operation code         */
        unsigned char   reserv2;            /* reserved                    */
        unsigned char   format;             /* format                      */
        unsigned short  primary_rc;         /* primary return code         */
        unsigned long   secondary_rc;       /* secondary return code       */
        unsigned char   *buf_ptr;           /* pointer to buffer           */
        unsigned long   buf_size;           /* buffer size                 */
        unsigned long   total_buf_size;     /* total buffer size required  */
        unsigned short  num_entries;        /* number of entries           */
        unsigned short  total_num_entries;  /* total number of entries     */
        unsigned char   list_options;       /* listing options             */
        unsigned char   reserv3;            /* reserved                    */
        unsigned char   resource_name[17];  /* network qualified res name  */
        unsigned char   reserv4;            /* reserved                    */
        unsigned short  resource_type;      /* Resource type               */
        unsigned char   parent_name[17];    /* parent name filter          */
        unsigned char   reserv5;            /* reserved                    */
        unsigned short  parent_type;        /* parent type                 */
} QUERY_DIRECTORY_ENTRY;
```

提供N数

＆C L序a供下P N }:

opcode

AP_QUERY_DIRECTORY_ENTRY

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。}K 0响 VCB D q =， v q = 1 返回 AP_DLUR_LU_RESOURCE D资源。

buf_ptr

指向C Z写入P m信息D缓e x D指k。＆C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ；，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m >无限制。

list_options

m w在P m信息中＆C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **resource_name** 和 **resource_type** D组合（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

resource_name

网g限定D资源{。C { 为 17 字Z $ R I EBCDIC U q R n d。| I = v A型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。(？v { F D $度最多I为 8 v字Z， " ；含6入U q。）如果 **list_options** h置为AP_FIRST_IN_LIST，则忽T C字段。

resource_type

资源类型。见下P值之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE
AP_DLUR_LU_RESOURCE

如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

parent_name

8代{ F过K器。{ F $度为 17 v字Z， "以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。(? v { F D $ 度最多I为 8 v字Z， " ；含6入U q。）如果h置K C字段，则v返回 t Z指定8代D目<项（ " R在b种 i v下，还X须h置 **parent_name** 字段）。如果C字段h置为全c，则忽T C字段。

parent_type

parent_name 字段中y指定D 8代D类型。如果 **parent_name** 字段为非c，则X须指定类型；否则C字段& h置为c。 C值I以h置为下P之一:

AP_ENCP_RESOURCE
AP_NNCP_RESOURCE

如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size

返回值m w返回 y P ； k s D P m信息y需要D缓e x D大小。 C值I以大Z **buf_size**。

num_entries

返回D目<项D }目。

total_num_entries

已 - 返回D项D总 }。 C值I以大Z **num_entries**。

directory_entry_summary.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

directory_entry_summary.resource_name

网g限定D资源{。{ F $度为 17 v字Z， "以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。(? v { F D $度最多I为 8 v字Z， " ；含6入U q。）

directory_entry_summary.resource_type

资源类型。 I能G下P之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE
AP_DLUR_LU_RESOURCE

（如果 **format** h置为c，则；返回。）

directory_entry_summary.description

资源5 w，如下y指定D:

DEFINE_LOCAL_LU
DEFINE_DIRECTORY_ENTRY
DEFINE_ADJACENT_LEN_NODE 或
DEFINE_ADJACENT_NODE

directory_entry_summary.real_owning_cp_type

ｖＣＺ NN 和 BrNN: ｆ｝５ＰＤ CP 类型。Ｉ能Ｇ下Ｐ之一:

AP_NONE

ｆ｝５ＰＤ CP Ｇ８代资源。

AP_ENCP_RESOURCE

ｆ｝５Ｐ CP ；Ｇ８代资源，而Ｇ一ｖ EN。

其｜Ｚｃ类型: Ｃ字段ｈ置为 AP_NONE。

directory_entry_summary.real_owning_cp_name

ｖＣＺ NN 和 BrNN: ｆ｝５Ｐ CP Ｄ全限定｛Ｆ。｛Ｆ＄度为 17 ｖ字Ｚ，"以 EBCDIC ＵｑＲｎｄ。Ｃ｛字ＧＩ＝ｖＣ EBCDIC ｃ串＊Ｄ Ａ型 EBCDIC 字符串构Ｉ。（？ｖ｛ＦＤ＄度最多Ｉ为 ８ ｖ字Ｚ，"；含６入Ｕｑ。）

如果ｆ｝５Ｐ CP Ｇ８代，则Ｃ字段ｈ置为二ｘ制ｃ。

如果ｆ｝５Ｐ CP ；Ｇ８代，则Ｃ字段ｈ置为ｆ｝５Ｐ CP Ｄ｛Ｆ。

如果 BrNN ｒ中Ｄ一ｖ EN ５Ｐ资源，则ｆ｝５Ｐ CP 在 BrNN Ｄ NNS Ｄ目＜中；Ｇ８代。在ｂ种ｉｖ下，ｆ｝５Ｐ CP Ｇ EN，＋８代Ｇ BrNN。

致ＢＢ一；

U q R n d 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。(? v { F D $ 度最多 I 为 8 v字 Z , " ; 含 6 入 U q 。)

directory_entry_detail.parent_type
8 代资源类型。 I 能 G 下 P 之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE

directory_lu_detail.entry_type
指定目 < 项 D 类型。 | I h 置为下 P 值之一:

AP_HOME
> X 资源。

AP_CACHE
_ Y 缓存项。

AP_REGISTER
注 a 资源 (v C Z NN)。

directory_entry_detail.location
指定资源 D 位置, I 以 G 下 P 值之一:

AP_LOCAL
资源在 > X Z c O。

AP_DOMAIN
资源 t Z , S D 端 Z c 。

AP_CROSS_DOMAIN
资源 ; 在 > X Z c D r 内。

directory_entry_detail.real_owning_cp_type
v C Z NN 和 BrNN: f } 5 P CP 类型。 I 能 G 下 P 之一:

AP_NONE
f } 5 P CP G 8 代资源。

AP_ENCP_RESOURCE
f } 5 P CP ; G 8 代资源, 而 G 一 v EN。

其 | Z c 类型: C 字段 h 置为 AP_NONE。

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_RES_NAME

AP_INVALID_RES_TYPE
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DIRECTORY_LU

QUERY_DIRECTORY_LU 从目 < }] b中返回 LU P m。 b些信息以 * 要 q = 或详细信息 q = D P m返回。如需P关某 v X定 LU D信息，或要获C几 v 『段』中D P m信息， & h 置 **lu_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽T C字段。 k N阅Z 12页D 『 i 询Z c 』，以获取关Z如何 9 C P m q = D 3 0知6。

C P m y] **lu_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度D { F x行 ASCII 词 d ` 辑排序（ y] IBM D 6611 APPN MIB 定序）。如果选中K AP_LIST_FROM_NEXT，则返回D P m 4 定义D次序（无 [指定D项G否存在）从下一项 * <。

注意: v现在目 <中D J C DLUS D LU 也 I C i 询返回。

VCB a 构

```
typedef struct query_directory_lu
{
        unsigned short  opcode;              /* verb operation code         */
        unsigned char   reserv2;             /* reserved                    */
        unsigned char   format;              /* format                      */
        unsigned short  primary_rc;          /* primary return code         */
        unsigned long   secondary_rc;        /* secondary return code       */
        unsigned char   *buf_ptr;            /* pointer to buffer           */
        unsigned long   buf_size;            /* buffer size                 */
        unsigned long   total_buf_size;      /* total buffer size required  */
        unsigned short  num_entries;         /* number of entries           */
        unsigned short  total_num_entries;   /* total number of entries     */
        unsigned char   list_options;        /* listing options             */
        unsigned char   reserv3;             /* reserved                    */
        unsigned char   lu_name[17];         /* network qualified LU name   */
} QUERY_DIRECTORY_LU;

typedef struct directory_lu_summary
{
        unsigned short  overlay_size;        /* size of this entry          */
        unsigned char   lu_name[17];         /* network qualified LU name   */
        unsigned char   description[RD_LEN]; /* resource description        */
} DIRECTORY_LU_SUMMARY;

typedef struct directory_lu_detail
{
        unsigned short  overlay_size;        /* size of this entry          */
        unsigned char   lu_name[17];         /* network qualified LU name   */
        unsigned char   description[RD_LEN]; /* resource description        */
        unsigned char   server_name[17];     /* network qualified           */
                                             /* server name                 */
        unsigned char   lu_owner_name[17];   /* network qualified           */
                                             /* LU owner name               */
        unsigned char   location;            /* Resource location           */
        unsigned char   entry_type;          /* Type of the directory entry */
        unsigned char   wild_card;           /* type of wildcard entry      */
        unsigned char   apparent_lu_owner_name[17];
                                             /* apparent LU owner name      */
        unsigned char   reserva[3];          /* reserved                    */
} DIRECTORY_LU_DETAIL;
```

提供N数

& C L序a供下P N }:

QUERY_DIRECTORY_LU

opcode

AP_QUERY_DIRECTORY_LU

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

buf_ptr

指向C Z 写入 P m信息D缓 e x D指 k。 &C L 序 I + }] m加 = VCB D末尾（X须 + **buf_ptr** h 置为 NULL）。

buf_size

y a 供D缓 e x D大小。返回D }] + ; ，过 b v 大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过 b v 值。 c 值m > 无限制。

list_options

m w在 P m信息中& C返回 2 4:

AP_SUMMARY

v 返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **lu_name**（见下 P N }）m > w引值， C w引值C Z 指定要返回D 5 际信息D起 < c 。

AP_FIRST_IN_LIST

忽 T w引值，返回 P m从 P m D Z 一项 * <。

AP_LIST_FROM_NEXT

返回 P m从 P m中 4 a 供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回 P m从w引值 y 指定D那项 * <。

lu_name

网 g 限定D LU { F。{ F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（? v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 入U q。）如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽 T C字段。

返回N数

如果动词 I 功执行，则L 序返回下 P N }:

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大 Z **buf_size**。

num_entries

返回D目<项D}目。

total_num_entries

已-返回D项D总}。C值I以大Z **num_entries**。

directory_lu_summary.overlay_size

C项中D字Z}，即=下一v返回项（如果PD话）D偏移?。

directory_lu_summary.lu_name

网g限定D LU {F。{F$度为 17 v字Z，"以 EBCDIC UqRnd。|I=v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(?v{FD$度最多I为 8 v字Z，"；含6入Uq。)

directory_lu_summary.description

资源5w（如在 DEFINE_LOCAL_LU 或 DEFINE_ADJACENT_NODE 中5wD)。|G一v以>XI显>字符集m>D 16 字Z字符串。yP 16 v字Z都P效。

directory_lu_detail.overlay_size

C项中D字Z}，即=下一v返回项（如果PD话）D偏移?。

directory_lu_detail.lu_name

网g限定D LU {F。{F$度为 17 v字Z，"以 EBCDIC UqRnd。|I=v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(?v{FD$度最多I为 8 v字Z，"；含6入Uq。)

directory_lu_detail.description

资源5w（如在 DEFINE_LOCAL_LU 或 DEFINE_ADJACENT_NODE 中5wD)。|G一v以>XI显>字符集m>D 16 字Z字符串。yP 16 v字Z都P效。

directory_lu_detail.server_name

服务Z LU DZcD网g限定{。{F$度为 17 v字Z，"以 EBCDIC UqRnd。|I=v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(?v{FD$度最多I为 8 v字Z，"；含6入Uq。)

directory_lu_detail.lu_owner_name

5P LU DZcD网g限定{。{F$度为 17 v字Z，"以 EBCDIC UqRnd。|I=v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(?v{FD$度最多I为 8 v字Z，"；含6入Uq。)

directory_lu_detail.location

指定资源D位置，I以G下P值之一:

AP_LOCAL

资源在>XZcO。

AP_DOMAIN

资源tZ，SD端Zc。

AP_CROSS_DOMAIN

资源；在>XZcDr内。

directory_lu_detail.entry_type

指定目<项D类型。|Ih置为下P值之一:

AP_HOME
> X资源。

AP_CACHE
_ Y缓存项。

AP_REGISTER
注 a 资源（ v C Z NN)。

directory_lu_detail.wild_card
指定 LU +匹配D通配符D类型。

AP_OTHER
LU 项D未知类型。

AP_EXPLICIT
完{ D **lu_name** + C Z定位C LU。

AP_PARTIAL_WILDCARD
只P **lu_name** D非U q ？分+ C Z定位C LU。

AP_FULL_WILDCARD
y P **lu_names** 都指向C LU。

directory_lu_detail.apprent_lu_owner_name
v C Z NN 和 BrNN: 全限定D m f D LU y P _ CP D { F。{ F $度为17 v字Z, "以 EBCDIC U q R n d。C { 字G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I。(？v { F D $度最多I为 8 v字Z, " ；含6人U q。)

如果m f D LU y P _ G f } LU y P _, 则C字段h置为二x制c。

如果m f D LU y P _ ；G f } y P _, 则C字段h置为m f D LU y P _ D { F。

如果 BrNN r中D一v EN 5 P资源, 则f } LU y P _在 BrNN D NNS D目<中；G m f D LU y P _。在b种i v下, f } LU y P _ G EN, + m f D y P _ G BrNN。

其| Z c类型: C字段h置为二x制c。

如果因N }错误而9动词；能执行, 则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LU_NAME

AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行, 则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行, 则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DIRECTORY_STATS

C动词ｖＪＣＺ通信服务器。

QUERY_DIRECTORY_STATS 返回目＜｝］ｂ统计信息。（M端Ｚｃ而言，＃ｔ＿Ｙ缓存信息D统计。）C动词ＩＣＺ估计网ｇ定位通信？D级ｐ。M网ｇＺｃ而言，C信息ＩＣＺｗ｛Ｚｃｕ＜化１Ｉ配置D目＜＿Ｙ缓存D大小。

VCB　a 构

```
typedef struct query_directory_stats
{
        unsigned short  opcode;            /* verb operation code          */
        unsigned chaT/F4ueserv2;code          /*   */
        unsigned chaT/F4formatcode;            /*   */
        unsigned short            /*/* code           */
        unsigned            /*/* co4e           */
        unsigned            /*/**/
        unsigned            /**/
        unsigned  /*   */
        unsigned          /*   */

                unsigned            /**/

                        unsigned            /*/**/
                        unsigned  /*/*/**/
                        unsigned            /*/**/
                        unsigned  /*/*/*           */
                        unsigned  /*/*/*           */
                        /*4t   */
```

cur_caches
t。

cur_home_entries
主项D１０｝目。

cur_reg_entries
注a项D１０｝目。

cur_directory_entries
１０目<中项D总｝。

cache_hits
t。

cache_misses
t。

in_locates
ＳＵＤ直S定位D｝？。

in_bcast_locates
ＳＵＤ广%定位D｝？。

out_locates
发MD直S定位D｝？。

out_bcast_locates
发MD广%定位D｝？。

not_found_locates
返回“未R=”D直S定位D｝？。

not_found_bcast_locates
返回“未R=”D广%定位D｝？。

locates_outstanding
未完I定位D１０｝？（|括直S定位和广%定位)。

如果因ＺｃＰ未启动而9动词；能执行，则L序返回下ＰＮ}:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下ＰＮ}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DLC

QUERY_DLC 返回 P 关在 Z c 定义 D DLC D信息 P m。C信息 I 『确定 }] 』（执行期间动，U集 D }] ）和『定义 }] 』（DEFINE_DLC O D & C L 序 y a 供 D }] ）构 I 。

b 些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关某 v X 定 DLC D信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **dlc_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C 字段。 k N 阅 Z 12 页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **dlc_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。

如果选中 K AP_LIST_FROM_NEXT，则返回 P m 4 定义 D 次序（无 [指定 D 项 G 否存在）从下一项 * < 。

VCB a 构

```
typedef struct query_dlc
{
        unsigned short  opcode;            /* verb operation code         */
        unsigned char   attributes;        /* ver attributes              */
        unsigned char   format;            /* format                      */
        unsigned short  primary_rc;        /* primary return code         */
        unsigned long   secondary_rc;      /* secondary return code       */
        unsigned char   *buf_ptr;          /* pointer to buffer           */
        unsigned long   buf_size;          /* buffer size                 */
        unsigned long   total_buf_size;    /* total buffer size required  */
        unsigned short  num_entries;       /* number of entries           */
        unsigned short  total_num_entries; /* total number of entries     */
        unsigned char   list_options;      /* listing options             */
        unsigned char   reserv3;           /* reserved                    */
        unsigned char   dlc_name[8];       /* name of DLC                 */
} QUERY_DLC;

typedef struct dlc_summary
{
        unsigned short  overlay_size;      /* size of this entry          */
        unsigned char   dlc_name[8];       /* name of DLC                 */
        unsigned char   description[RD_LEN];
                                           /* resource description        */
        unsigned char   state;             /* State of the DLC            */
        unsigned char   dlc_type;          /* DLC type                    */
} DLC_SUMMARY;

typedef struct dlc_detail
{
        unsigned short  overlay_size;      /* size of this entry          */
        unsigned char   dlc_name[8];       /* name of DLC                 */
        unsigned char   reserv2[2];        /* reserved                    */
        DLC_DET_DATA     det_data;         /* Determined data             */
        DLC_DEF_DATA     def_data;         /* Defined data                */
} DLC_DETAIL;

typedef struct dlc_det_data
{
        unsigned char   state;             /* State of the DLC            */
        unsigned char   reserv3[3];        /* reserved                    */
        unsigned char   reserva[20];       /* reserved                    */
} DLC_DET_DATA;
```

```
typedef struct dlc_def_data
{
        DESCRIPTION     description;       /* resource description        */
        unsigned char   dlc_type;          /* DLC type                    */
        unsigned char   neg_ls_supp;       /* negotiable LS support       */
        unsigned char   port_types;        /* allowable port types        */
        unsigned char   retry_flags;       /* conditions for automatic    */
                                           /* retries                     */
        unsigned short  max_activaion_attempts;
                                           /* how many automatic retries? */
        unsigned short  activation_delay_timer;
                                           /* delay between automatic     */
                                           /* retries                     */
        unsigned char   reserv3[6];        /* reserved                    */
        unsigned short  dlc_spec_data_len; /* Length of DLC specific data */
} DLC_DEF_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_DLC

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **dlc_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT
返回 P m从 P m中 4 a 供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE
返回 P m从w引值 y 指定D那项 * <。

dlc_name
DLC { F。 | G在> X I 显>字符集中D一v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须h置。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词 I 功执行，则L序返回下 P N }:

primary_rc
AP_OK

buf_size
在缓 e x 中返回D信息 $ 度。

total_buf_size
返回值m w返回 y P ； k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。 C值 I 以大Z **num_entries**。

dlc_summary.overlay_size
C项中D字Z }， 即 = 下一 v 返回项（如果 P D话）D偏移 ?。

dlc_summary.dlc_name
DLC { F。 | G在> X I 显>字符集中D一v 8 字Z字符串。 y P D 8 v字Z G P效D。

dlc_summary.description
资源 5 w（如在 DEFINE_DLC 中 5 w D）。 | G一 v 以> X I 显>字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

dlc_summary.state
DLC D状，。 C字段 h 置为下 P 值之一:

AP_ACTIVE
AP_NOT_ACTIVE
AP_PENDING_INACTIVE

dlc_summary.dlc_type
DLC D类型。 v 人通信或通信服务器支V下 P 类型:

AP_ANYNET
AP_LLC2
AP_OEM_DLC

AP_SDLC
AP_TWINAX
AP_X25

dlc_detail.overlay_size
C项中D字Z }（ | 括 dlc_spec_data），即 = 下一 v 返回项（如果 P D话）D偏移？。

dlc_detail.dlc_name
DLC { F。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P效D。

dlc_detail.det_data.state
DLC D状，。 C字段 h 置为下 P 值之一:

AP_ACTIVE
AP_NOT_ACTIVE
AP_PENDING_INACTIVE

dlc_detail.def_data.description
资源 5 w（如在 DEFINE_DLC 中 5 w D）。 | G一 v 以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

dlc_detail.def_data.dlc_type
DLC D类型。 v 人通信或通信服务器支V下 P 类型:

AP_ANYNET
AP_LLC2
AP_OEM_DLC
AP_SDLC
AP_TWINAX
AP_X25

dlc_detail.def_data.neg_ls_supp
指定 DLC G否支V I 协L 4 7 >（AP_YES 或 AP_NO）。

dlc_detail.def_data.port_types
指定 y a 供D **dlc_type** D允许端Z类型。其值对& Z 下 P w值D OR（或Y作）组合中D一 v 或多 v:

AP_PORT_NONSWITCHED
AP_PORT_SWITCHED
AP_PORT_SATF

dlc__detail.def_data.retry_flags
C字段指定如果在 **def_data.retry_flags** D DEFINE_LS 和 DEFINE_PORT O h 置 AP_INHERIT_RETRY j 志，则C DLC O定义D 4 7 > I 自动重T D u件。 | G一 v 位字段，" I 9 C任何下P 4位 OR（或Y作）组合。

AP_RETRY_ON_START
在 " T激活 1，如果未从远L Z c S U = 响&，则重T 4 7 激活。如果在 " T激活 1，基 > 端Z G非活动D，则L序 + " T激活 |。

AP_RETRY_ON_FAILURE
如果在活动和暂挂活动 1 4 S ' \，则重 T 4 7 激活。如果在 " T 激活 1，基 > 端 Z 发 z 故 O，则 L 序 + " T 激活 | 。

AP_RETRY_ON_DISCONNECT
如果 4 7 I 远 L Z c 以 } # 方 = 停止，则重 T 4 7 激活。

AP_DELAY_APPLICATION_RETRIES
I & C L 序启动 D 4 7 激活重 T（ 9 C START_LS 或 k s = 4 7 激活）+ 9 C **activation_delay_timer** x 行 w = 。

AP_INHERIT_RETRY
C j 志无作 C。

dlc_detail.def_data.max_activation_attempts
C 字段；z z 效果，} 非在 **def_data.retry_flags** 中 D DEFINE_LS 中至 Y h 置一 v j 志，DEFINE_LS O D **def_data.max_activation_attempts** h 置为 AP_USE_DEFAULTS，以及 DEFINE_PORT O D **def_data.max_activation_attempts** h 置为 AP_USE_DEFAULTS。

C 字段指定 1 远 L Z c 无响 & 或基 > 端 Z ；活动 1，L 序允许重 T D 次 } 。 | | 括自动重 T 和 I & C L 序 } 动 D 激活 " T = _ 。

如果达 = C 极限，则；再 x 行自动重 T。C u 件 I STOP_LS、STOP_PORT、STOP_DLC 或一 v I 功激活 4 位。START_LS 或 OPEN_LU_SSCP_SEC_RQ z z 一 v % v 激活 " T，| 在激活 ' \ 1 ；再重 T。

c m > '无限制'。值 AP_USE_DEFAULTS m > '无限制'。

dlc_detail.def_data.activation_delay_timer
C 字段；z z 效果，} 非在 **def_data.retry_flags** 中 D DEFINE_LS 中至 Y h 置一 v j 志，DEFINE_LS O D **def_data.max_activation_attempts** h 置为 AP_USE_DEFAULTS，以及 DEFINE_PORT O D **def_data.max_activation_attempts** h 置为 AP_USE_DEFAULTS。

C 字段指定在自动重 T 之间 L 序 H 待 D k } ，以及如果在 **def_data.retry_flags** 中 h 置 AP_DELAY_APPLICATION_RETRIES 位，I & C L 序 } 动 D 激活 " T 之间 L 序 H 待 D k } 。

c 值或 AP_USE_DEFAULTS 9 C 缺 ! 计 1 器值 30 k 。

dlc_detail.def_data.dlc_spec_data_len
X 定 Z DLC 类型 D }] D 未 m z $ 度（以字 Z 为 % 位）。}] + k DLC_DETAIL a 构，S 。C }] + n z 4 字 Z 范围。C 字段 & < 终 h 置为 c 。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DLC_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DLUR_DEFAULTS

QUERY_DLUR_DEFAULTS 9 C户能 i 询C DEFINE_DLUR_DEFAULTS 动词 y 定义 D缺！值。

VCB a 构

```
typedef struct query_dlur_defaults
{
        unsigned short  opcode;            /* verb operation code         */
        unsigned char   reserv2;           /* reserved                    */
        unsigned char   format;            /* format                      */
        unsigned short  primary_rc;        /* primary return code         */
        unsigned long   secondary_rc;      /* secondary return code       */
        DESCRIPTION     description;       /* resource description        */
        unsigned char   dlus_name[17];     /* DLUS name                    */
        unsigned char   bkup_dlus_name[17];/* Backup DLUS name              */
        unsigned char   reserv3;           /* reserved                    */
        unsigned short  dlus_retry timeout;/* DLUS Retry Timeout          */
        unsigned short  dlus_retry_limit;  /* DLUS Retry Limit            */
        unsigned char   reserv4[16];       /* reserved                    */
} QUERY_DLUR_LU;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_DLUR_LU

format

j 6 VCB D q =。 + C字段 h 置为 c，以指定O f P v D VCB D f >。

返回N数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

description

资源 5 w。 C字段D $ 度& 为 4 v 字 Z D 6 }， R ；为 c。

dlus_name

+ 作为缺！值 9 C D DLUS Z c D { F。 | I 以 h 置为 c，或 17 字 Z D字符串。 C字符串 I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串组 I， R I EBCDIC U q R n d。（ ? v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人U q。）

bkup_dlus_name

+ 作为 8 份缺！值 9 C D DLUS Z c D { F。 | I 以 h 置为 c，或 17 字 Z D 字符串。 C字符串 I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串组 I， R I EBCDIC U q R n d。（ ? v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人 U q。） LU D位置。 v 返回下 P 值:

dlus_retry_timeout

在 Z 二次 " T k DLUS * 系和后继 " T 之间 D 间 t （ k ）。 u < " T 和 Z 一次重 T 之间 D 间 t < 终为 1 k 。

dlus_retry_limit

在 Z 一次 k DLUS * 系 ' \ 后最大 D 重 T 次 } 。如果指定 X'FFFF'，则 L 序 + ；确定 X x 行重 T 。

如果因相关 START_NODE N } 未 h 置而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果因未 (" DLUR 支 V D 系统而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_INVALID_VERB

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因 STOP_NODE 动词 P 未发 v 而 9 动词；能执行， L 序则返回下 P N } :

primary_rc

AP_NODE_STOPPING

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DLUR_LU

QUERY_DLUR_LU 返回 P 关 DLUR 支 V D LU D 信息。

b 些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关某 v X定 LU D信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **lu_name** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。 k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **lu_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。

如果选中 K AP_LIST_FROM_NEXT，则返回 D P m 4 定义 D 次序（无 [指定 D 项 G 否存在）从下一项 * < 。

返回 D LU P m I 以 y] **pu_name**，或 y] LU G > X D还 G 下 N D，或 y] b = _ 来过 K。如果希望 y] PU 过 K，则 h 置 **pu_name** 字段（否则，C字段 h 置为全 c ）。如果希望 y] 位置过 K，则 **filter** 字段 & h 置为 AP_INTERNAL 或 AP_DOWNSTREAM（否则，若；需要过 K，则 C 字段 & h 置为 AP_NONE）。

VCB a 构

```
typedef struct query_dlur_lu
{
        unsigned short  opcode;            /* verb operation code         */
        unsigned char   reserv2;           /* reserved                    */
        unsigned char   format;            /* format                      */
        unsigned short  primary_rc;        /* primary return code         */
        unsigned long   secondary_rc;      /* secondary return code       */
        unsigned char   *buf_ptr;          /* pointer to buffer           */
        unsigned long   buf_size;          /* buffer size                 */
        unsigned long   total_buf_size;    /* total buffer size required  */
        unsigned short  num_entries;       /* number of entries           */
        unsigned short  total_num_entries; /* total number of entries     */
        unsigned char   list_options;      /* listing options             */
        unsigned char   reserv3;           /* reserved                    */
        unsigned char   lu_name[8];        /* name of LU                  */
        unsigned char   pu_name[8];        /* name of PU to filter on     */
        unsigned char   filter;            /* reserved                    */
} QUERY_DLUR_LU;

typedef struct dlur_lu_summary
{
        unsigned short  overlay_size;      /* size of this entry          */
        unsigned char   lu_name[8];        /* name of LU                  */
} DLUR_LU_SUMMARY;

typedef struct dlur_lu_detail
{
        unsigned short  overlay_size;      /* size of this entry          */
        unsigned char   lu_name[8];        /* name of LU                  */
        unsigned char   pu_name[8];        /* name of owning PU           */
        unsigned char   dlus_name[17];     /* DLUS name if SSCP-LU        */
                                           /* session active              */
        unsigned char   lu_location;       /* downstream or local LU      */
        unsigned char   nau_address;       /* NAU address of LU           */
        unsigned char   plu_name[17];      /* PLU name if PLU-SLU session */
```

```
                                        /* active                       */
        unsigned char   reserv1[27];    /* reserved                     */
        unsigned char   rscv_len;       /* length of appended RSCV      */
} DLUR_LU_DETAIL;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_DLUR_LU

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ?；要，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **lu_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

lu_name

} i询D LU D { F。b G v 8字Z D字母}字集D A型 EBCDIC 字符串（以字母 * <）， I EBCDIC U q R n d。如果 **list_options** h置为AP_FIRST_IN_LIST，则忽T C字段。

pu_name

PU { F过K器。|&C h置为全c或为一v 8字Z字母}字D A型 EBCDIC字符串（以一v字母 * 头）， " 以 EBCDIC U q R n d。如果h置K C字段，则只返回k指定 PU 相关D LU。 如果C字段h置为全c，则忽T C字段。

filter 位置过K器。指定G否& y]位置（AP_INTERNAL 或 AP_DOWNSTREAM）过K返回D LU。如果；需要过K，则C字段& h置为 AP_NONE。

QUERY_DLUR_LU

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**。

dlur_lu_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

dlur_lu_summary.lu_name
LU D { F。b G一v 8字Z字母 }字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC U q R n d。

dlur_lu_detail.overlay_size
C项中D字Z }（|括任何=加RSCV），即=下一v返回项（如果P D话）D偏移?。

dlur_lu_detail.lu_name
LU D { F。b G一v 8字Z字母 }字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC U q R n d。

dlur_lu_detail.pu_name
k LU关*D PU D { F。b G一v 8字Z字母 }字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC U q R n d。

dlur_lu_detail.dlus_name
如果SSCP_LU会话G活动D，则|G DLUS Z c D { F。|G一v 17字Z D字符串，I = v C EBCDIC c串*D A型EBCDIC字符串组I，R I EBCDIC U q R n d。(?v { F D $度最多I为8 v字Z，"；含6入U q。）如果SSCP-LU会话；G活动D，则C字段+h置为全c。

dlur_lu_detail.lu_location
LU D位置。v返回下P值:

AP_INTERNAL
AP_DOWNSTREAM

dlur_lu_detail.nau_address
LU D网g I访问%元X址。范围在1 -- 255之间。

dlur_lu_detail.plu_name
如果LU P活动D PLU-SLU会话，则|G PLU D { F。|G一v 17字Z D字符串，I = v C EBCDIC c串*D A型EBCDIC字符串组I，R I

EBCDIC Uq R n d。(？v { F D $度最多I为 8 v字Z, " ；含6入U q。）如果 PLU-SLU 会话；G活动D，则C字段+ h置为全c。

dlur_lu_detail.rscv_len

C值+ <终为c。

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_INVALID_FILTER_OPTION
AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DLUR_PU

QUERY_DLUR_PU 返回 P 关 DLUR 支 V D PU D信息 P m。

b 些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关某 v X定 PU D信息，或要获 C 几 v 『段』中 D P m信息，& h 置 **pu_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **pu_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。

如果选中 K AP_LIST_FROM_NEXT，则返回 D P m 4 定义 D次序（无 [指定 D项 G 否存在）从下一项 * < 。

返回 D PU P m I 以 y] **dlus_name**，或 y] PU G > X D还 G 下 N D，或 y] b = _ 来过 K。如果希望 y] DLUS 过 K，则 h 置 **dlus_name** 字段（否则，C字段 h 置为全 c ）。如果希望 y] PU 位置过 K，则 **filter** 字段 & h 置为 AP_INTERNAL 或 AP_DOWNSTREAM（否则，若；需要过 K，则 C 字段 & h 置为 AP_NONE）。

VCB a 构

```
typedef struct query_dlur_pu
{
        unsigned short  opcode;              /* verb operation code           */
        unsigned char   reserv2;             /* reserved                      */
        unsigned char   format;              /* format                        */
        unsigned short  primary_rc;          /* primary return code           */
        unsigned long   secondary_rc;        /* secondary return code         */
        unsigned char   *buf_ptr;            /* pointer to buffer             */
        unsigned long   buf_size;            /* buffer size                   */
        unsigned long   total_buf_size;      /* total buffer size required    */
        unsigned short  num_entries;         /* number of entries             */
        unsigned short  total_num_entries;   /* total number of entries       */
        unsigned char   list_options;        /* listing options               */
        unsigned char   reserv3;             /* reserved                      */
        unsigned char   pu_name[8];          /* name of PU                    */
        unsigned char   dlus_name[17];       /* fully qualified DLUS name     */
        unsigned char   filter;              /* local/downstream filter       */
} QUERY_DLUR_PU;

typedef struct dlur_pu_summary
{
        unsigned short  overlay_size;        /* size of this entry            */
        unsigned char   pu_name[8];          /* name of PU                    */
        unsigned char   description[RD_LEN];
                                             /* resource description          */
} DLUR_PU_SUMMARY;

typedef struct dlur_pu_detail
{
        unsigned short  overlay_size;        /* size of this entry            */
        unsigned char   pu_name[8];          /* name of PU                    */
        unsigned char   description[RD_LEN];
                                             /* resource description          */
        unsigned char   defined_dlus_name[17];
                                             /* defined DLUS name             */
        unsigned char   bkup_dlus_name[17];  /* backup DLUS name              */
        unsigned char   pu_id[4];            /* PU identifier                 */
        unsigned char   pu_location;         /* downstream or local PU        */
        unsigned char   active_dlus_name[17];
```

```
                                        /* active DLUS name              */
        unsigned char   ans_support;    /* Auto-Network shutdown support */
        unsigned char   pu_status;      /* status of the PU              */
        unsigned char   dlus_session_status; /* status of the DLUS pipe  */
        unsigned char   reserv3;        /* reserved                      */
        FQPCID  fqpcid;                 /* FQPCID used on pipe           */
        unsigned short  dlus_retry_timeout; /* DLUS retry timeout        */
        unsigned short  dlus_retry_limit;   /* DLUS retry limit          */
} DLUR_PU_DETAIL;

typedef struct fqpcid
{
        unsigned char   pcid[8];        /* proc correlator identifier    */
        unsigned char   fqcp_name[17];  /* originator's network          */
                                        /* qualified CP name             */
        unsigned char   reserve3[3];    /* reserved                      */
} FQPCID;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_DLUR_PU

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。 &C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **pu_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

QUERY_DLUR_PU

pu_name

} i 询D PU D { F。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以字母 * < ），I EBCDIC U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

dlus_name

DLUS 过K器。C值& h置为全c，或I = v A 型 EBCDIC 字符串组I、中间以一v EBCDIC c ，S、 " 以 EBCDIC U q R n d D 17 字Z字符串。如果h置K C字段，那4只返回k指定 DLUS Z c相，D SSCP-PU 会话相关D PU。 如果C字段h置为全c，则忽T C字段。

filter C字段& h置为 AP_NONE。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size

返回值m w返回y P ；k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总 }。C值I以大Z **num_entries**。

dlur_pu_summary.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

dlur_pu_summary.pu_name

PU D { F。 b G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v字母 * 头），" 以 EBCDIC U q R n d。

dlur_pu_summary.description

资源5 w（如在 DEFINE_INTERNAL_PU 中5 w D）。 | G一v以> X I显> 字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

dlur_pu_detail.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

dlur_pu_detail.pu_name

PU D { F。 b G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v字母 * 头），" 以 EBCDIC U q R n d。

dlur_pu_detail.description

资源5 w（如在 DEFINE_INTERNAL_PU 中5 w D）。 | G一v以> X I显> 字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

dlur_pu_detail.defined_dlus_name

I DEFINE_INTERNAL_PU 动词或 DEFINE_LS 动词（R **dspu_services** h

置为 AP_DLUR）y 定义D DLUS Z c D { F。 | G一v 17 字Z D字符串, I = v C EBCDIC c串 * D A 型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（？v { F D $ 度最多I为 8 v字Z， " ；含6人U q。）

dlur_pu_detail.bkup_dlus_name

I DEFINE_INTERNAL_PU 动词或 DEFINE_LS 动词（R **dspu_services** h 置为 AP_DLUR）y 定义D 8 份 DLUS Z c D { F。 | G一v 17 字Z D字符串， I = v C EBCDIC c串 * D A 型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（？v { F D $ 度最多I为 8 v字Z， " ；含6人U q。）

dlur_pu_detail.pu_id

在 DEFINE_INTERNAL_PU 动词中定义D或从下N PU D XID 获C D PU j 6符。 | G一v 4 字Z . y x制字符串。位 0--11 h置为i号，而位 12--31 置为唯一6 p PU D ID 号。

dlur_pu_detail.pu_location

QUERY_DLUR_PU

dlur_pu_detail.dlus_session_status
1 0 PU } 9 C D DLUS 管@D状，。 | I h置为下P值之一:

AP_PENDING_ACTIVE
AP_ACTIVE
AP_PENDING_INACTIVE
AP_INACTIVE

dlur_pu_detail.fqpcid.pcid
管@O 9 C D过L相关因子 ID。 b G v 8 字Z D . y x制字符串。如果SSCP-PU 会话；G活动D，则h置C字段为c。

dlur_pu_detail.fqpcid.fqcp_name
管@O 9 C D全限定X制c { F。 { F $度为 17 v字Z， "以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c , S。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）如果 SSCP-PU 会话；G活动D，则h置C字段为c。

dlur_pu_detail.dlus_retry_timeout
在Z二次 " T k在 **dlus_name** 和 **bkup_dlus_name** 字段中指定D DLUS * 系和后继 " T之间D间 t （ k）。 u < " T和Z一次重T之间D间 t <终为 1 k。如果指定为c，则+ 9 C通过 DEFINE_DLUR_DEFAULTS 配置D缺！值。

def_data.dlus_retry_limit
在Z一次k在 **dlus_name** 和 **bkup_dlus_name** 字段中指定D DLUS *系 ' \ 之后，最多D重T次 }。如果指定为c，则+ 9 C通过 DEFINE_DLUR_DEFAULTS 配置D缺！值。如果指定 X'FFFF'，则L序I以无 }次重T。

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_PU_NAME

AP_INVALID_FILTER_OPTION
AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DLUS

QUERY_DLUS 返回 P 关 DLUR y 知 D DLUS D信息 P m。

C信息作为 P m返回。如需 P 关某 v X定 DLUS D信息，或要获 C 几 v 『段』中 D P m 信息， & h 置 **dlus_name** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。 k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **dlus_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。

如果选中 K AP_LIST_FROM_NEXT，则返回 D P m 4 定义 D 次序（无 [指定 D 项 G 否存在）从下一项 * < 。

注意: C动词返回管@统计信息。

VCB a 构

```
typedef struct query_dlus
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   *buf_ptr;          /* pointer to buffer             */
        unsigned long   buf_size;          /* buffer size                   */
        unsigned long   total_buf_size;    /* total buffer size required    */
        unsigned short  num_entries;       /* number of entries             */
        unsigned short  total_num_entries; /* total number of entries       */
        unsigned char   list_options;      /* listing options               */
        unsigned char   reserv3;           /* reserved                      */
        unsigned char   dlus_name[17];     /* fully qualified DLUS name     */
} QUERY_DLUS;

typedef struct dlus_data
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   dlus_name[17];     /* fully qualified DLUS name     */
        unsigned char   is_default;        /* is the DLUS the default       */
        unsigned char   is_backup_default; /* is DLUS the backup default    */
        unsigned char   pipe_state;        /* state of CPSVRMGR pipe        */
        unsigned short  num_active_pus;    /* num of active PUs using pipe  */
        PIPE_STATS      pipe_stats;        /* pipe statistics               */
} DLUS_DATA;

typedef struct pipe_stats
{
        unsigned long   reqactpu_sent;     /* REQACTPUs sent to DLUS        */
        unsigned long   reqactpu_rsp_received;
                                           /* RSP(REQACTPU)s received       */
                                           /* from DLUS                     */
        unsigned long   actpu_received;    /* ACTPUs received from DLUS     */
        unsigned long   actpu_rsp_sent;    /* RSP(ACTPU)s sent to DLUS      */
        unsigned long   reqdactpu_sent;    /* REQDACTPUs sent to DLUS       */
        unsigned long   reqdactpu_rsp_received;
                                           /* RSP(REQDACTPU)s received      */
                                           /* from DLUS                     */
        unsigned long   dactpu_received;   /* DACTPUs received from DLUS    */
        unsigned long   dactpu_rsp_sent;   /* RSP(DACTPU)s sent to DLUS     */
```

```
        unsigned long   actlu_received;    /* ACTLUs received from DLUS     */
        unsigned long   actlu_rsp_sent;    /* RSP(ACTLU)s sent to DLUS      */
        unsigned long   dactlu_received;   /* DACTLUs received from DLUS    */
        unsigned long   dactlu_rsp_sent;   /* RSP(DACTLU)s sent to DLUS     */
        unsigned long   sscp_pu_mus_rcvd;  /* MUs for SSCP-PU               */
                                           /* sessions received             */
        unsigned long   sscp_pu_mus_sent;  /* MUs for SSCP-PU sessions sent */
        unsigned long   sscp_lu_mus_rcvd;  /* MUs for SSCP-LU sessions      */
                                           /* received                      */
        unsigned long   sscp_lu_mus_sent;  /* MUs for SSCP-LU sessions sent */
} PIPE_STATS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_DLUS

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过C大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **dlus_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

dlus_name

} i询D DLUS D { F。C值& h置为全c，或 17 字Z字符串。| I = v A型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S， " 以 EBCDIC U q R n d。(? v { F D $ 度最多I为 8 v字Z， " ；含6入U q。）如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息$度。#

APX_OK

dlus_data.pipe_stats.reqdactpu_sent
通过管@发M x DLUS D REQDACTPU D } ?。

dlus_data.pipe_stats.reqdactpu_rsp_received
通过管@从 DLUS S U D RSP(REQDACTPU) D } ?。

dlus_data.pipe_stats.dactpu_received
通过管@从 DLUS S U D DACTPU D } ?。

dlus_data.pipe_stats.dactpu_rsp_sent
通过管@发M x DLUS D RSP(DACTPU) D } ?。

dlus_data.pipe_stats.actlu_received
通过管@从 DLUS S U D ACTLU D } ?。

dlus_data.pipe_stats.actlu_rsp_sent
通过管@发M x DLUS D RSP(ACTLU) D } ?。

dlus_data.pipe_stats.dactlu_received
通过管@从 DLUS S U D DACTLU D } ?。

dlus_data.pipe_stats.dactlu_rsp_sent
通过管@发M x DLUS D RSP(DACTLU) D } ?。

dlus_data.pipe_stats.sscp_pu_mus_rcvd
通过管@从 DLUS S U D SSCP-PU MU D } ?。

dlus_data.pipe_stats.sscp_pu_mus_sent
通过管@发M x DLUS D SSCP-PU MU D } ?。

dlus_data.pipe_stats.sscp_lu_mus_rcvd
通过管@从 DLUS S U D SSCP-LU MU D } ?。

dlus_data.pipe_stats.sscp_lu_mus_sent
通过管@发 x DLUS D SSCP-LU MU D } ?。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DLUS_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DOWNSTREAM_LU

C动词v J C Z通信服务器。

QUERY_DOWNSTREAM_LU 返回P关I DLUR 或 PU 集中或b = _兼P服务D下N LU D信息。C信息I确定}]（执行期间动，U集D }]）和定义}]构I。（定义}] I DEFINE_DOWNSTREAM_LU 动词O D & C L序y a供。注意: 对Z DLUR 支V D LU，在激活下N LU 1，隐=定义}]M位。）

b些信息以*要q =或详细信息q = D P m返回。如需P关某v X定> X LU D信息，或要获C几v段中D P m信息，& h置 **dslu_name** 字段。否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽T C字段。

返回D LU I以y] > X Z c y a供D服务类型或 LU D关*下N PU 或b = _兼P来过K。如果希望y]服务类型过K，则 **dspu_services** 字段& h置为 AP_PU_CONCENTRATION 或 AP_DLUR（否则，C字段& h置为 AP_NONE）。如果希望y] PU 过K，则& h置 **dspu_name** 字段（否则，C字段& h置为全c）。

VCB a构

```
typedef struct query_downstream_lu
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   attributes;        /* Verb attributes               */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   *buf_ptr;          /* pointer to buffer             */
        unsigned long   buf_size;          /* buffer size                   */
        unsigned long   total_buf_size;    /* total buffer size required    */
        unsigned short  num_entries;       /* number of entries             */
        unsigned short  total_num_entries; /* total number of entries       */
        unsigned char   list_options;      /* listing options               */
        unsigned char   reserv3;           /* reserved                      */
        unsigned char   dslu_name[8];      /* Downstream LU name            */
        unsigned char   dspu_name[8];      /* Downstream PU name filter     */
        unsigned char   dspu_services;     /* filter on DSPU services type  */
} QUERY_DOWNSTREAM_LU;

typedef struct downstream_lu_summary
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   dslu_name[8];      /* LU name                       */
        unsigned char   dspu_name[8];      /* PU name                       */
        unsigned char   description[RD_LEN];
                                           /* resource description          */
        unsigned char   dspu_services;     /* type of service provided to   */
                                           /* downstream node               */
        unsigned char   nau_address;       /* NAU address                   */
        unsigned char   lu_sscp_sess_active;
                                           /* Is LU-SSCP session active     */
        unsigned char   plu_sess_active;   /* Is PLU-SLU session active     */
} DOWNSTREAM_LU_SUMMARY

typedef struct downstream_lu_detail
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   dslu_name[8];      /* LU name                       */
```

```
        unsigned char   reserv1[2];        /* reserved                      */
        DOWNSTREAM_LU_DET_DATA det_data;   /* Determined data               */
        DOWNSTREAM_LU_DEF_DATA def_data;   /* Defined data                  */
} DOWNSTREAM_LU_DETAIL;

typedef struct downstream_lu_det_data
{
        unsigned char   lu_sscp_sess_active;
                                           /* Is LU-SSCP session active     */
        unsigned char   plu_sess_active;   /* Is PLU-SLU session active     */
        unsigned char   dspu_services;     /* type of services provided to  */
                                           /* downstream node               */
        unsigned char   reserv1;           /* reserved                      */
        SESSION_STATS   lu_sscp_stats;     /* LU-SSCP session statistics    */
        SESSION_STATS   ds_plu_stats;      /* downstream PLU-SLU session    */
                                           /* statistics                    */
        SESSION_STATS   us_plu_stats;      /* upstream PLU_SLU sess stats   */
        unsigned char   host_lu_name[8];   /* Determined host LU name       */
        unsigned char   host_pu_name[8];   /* Determined host PU name       */
        unsigned char   reserva[4];        /* reserved                      */
} DOWNSTREAM_LU_DET_DATA;

typedef struct downstream_lu_def_data
{
        unsigned char   description[RD_LEN];
                                           /* resource description          */
        unsigned char   nau_address;       /* NAU address                   */
        unsigned char   dspu_name[8];      /* Downstream PU name            */
        unsigned char   host_pu_name;      /* host LU or pool name          */
        unsigned char   allow_timeout;     /* Allow timeout of host LU?     */
        unsigned char   delayed_logon;     /* Allow delayed logo to host LU */
        unsigned char   reserv2[6];        /* reserved                      */
} DOWNSTREAM_LU_DEF_DATA

typedef struct session_stats
{
        unsigned short  rcv_ru_size;       /* session receive RU size       */
        unsigned short  send_ru_size;      /* session send RU size          */
        unsigned short  max_send_btu_size; /* max send BTU size             */
        unsigned short  max_rcv_btu_size;  /* max rcv BTU size              */
        unsigned short  max_send_pac_win;  /* max send pacing win size      */
        unsigned short  cur_send_pac_win;  /* current send pacing win size  */
        unsigned short  max_rcv_pac_win;   /* max receive pacing win size   */
        unsigned short  cur_rcv_pac_win;   /* current receive pacing        */
                                           /* window size                   */
        unsigned long   send_data_frames;  /* number of data frames sent    */
        unsigned long   send_fmd_data_frames;
                                           /* num of FMD data frames sent   */
        unsigned long   send_data_bytes;   /* number of data bytes sent     */
        unsigned long   rcv_data_frames;   /* num data frames received      */
        unsigned long   rcv_fmd_data_frames;
                                           /* num of FMD data frames recvd  */
        unsigned long   rcv_data_bytes;    /* number of data bytes received */
        unsigned char   sidh;              /* session ID high byte          */
        unsigned char   sidl;              /* session ID low byte           */
        unsigned char   odai;              /* ODAI bit set                  */
        unsigned char   ls_name[8];        /* Link station name             */
        unsigned char   pacing type;       /* type of pacing in use         */
} SESSION_STATS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_DOWNSTREAM_LU

attributes

动词D t 性。C字段G一 v 位字段。Z 一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。

buf_size

y a供D缓e x D大小。返回D }] + ；，过C大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过b v 值。 c 值m > 无限制。

list_options

m w在P m信息中& C返回 2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **dslu_name**（见下P N } ）m > w引值， C w引值C Z指定要返回D 5际信息D起 < c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z 一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中 4 a供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值 y 指定D那项 * <。

dslu_name

} i 询D > X LU D { F。b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

dspu_name

PU { F过K器。| & C h置为全 c 或为一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。如果 h 置K C字段，则只返回 k 指定 PU 相关D LU。如果C字段 h 置为全 c，则忽T C字段。

dspu_services

DSPU 服务过K器。如果 h 置为 AP_PU_CONCENTRATION，则 v 返回 PU 集中服务D下N LU。如果 h 置为 AP_DLUR，则 v 返回 DLUR 支V D LU。否则，如果 h 置为 AP_NONE，则返回 y P下N LU D信息。

QUERY_DOWNSTREAM_LU

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总}。C值I以大Z **num_entries**。

downstream_lu_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

downstream_lu_summary.dslu_name
} i 询D > X LU D { F。b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。

downstream_lu_summary.dspu_name
C LU } 9 C D > X PU D { F。b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。

downstream_lu_summary.description
资源5 w（如在 DEFINE_DOWNSTREAM_LU 或 DEFINE_DOWNSTREAM_LU_RANGE 中5 w D）。| G一v以>X I显>字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

downstream_lu_summary.dspu_services
指定>X Z c a供x下N LU g越4 7 D服务。| h置为下P之一:

AP_PU_CONCENTRATION
向下N LU a供 PU 集中D > X Z c。

AP_DLUR
向下N LU a供 DLUR 支V D > X Z c。

downstream_lu_summary.nau_address
LU D网g I寻址%元X址，范围在 1--255 内。

downstream_lu_summary.lu_sscp_sess_active
指> LU-SSCP 会话G否G活动D（AP_YES 或 AP_NO）。

downstream_lu_summary.plu_sess_active
指> PLU-SLU 会话G否G活动D（AP_YES 或 AP_NO）。

downstream_lu_detail.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

downstream_lu_detail.dslu_name
} i 询D > X LU D { F。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。

downstream_lu_detail.det_data.lu_sscp_sess_active
指 > = 下N LU D LU-SSCP 会话G否G活动D（AP_YES 或 AP_NO）。

downstream_lu_detail.det_data.plu_sess_active
指 > = 下N LU D PLU-SLU 会话G否G活动D（AP_YES 或 AP_NO）。

downstream_lu_detail.det_data.dspu_services
指定 > X Z c a 供 x 下N LU g 越 4 7 D服务。 | h 置为下 P 值之一:

AP_PU_CONCENTRATION
向下N LU a 供 PU 集中D > X Z c。

AP_DLUR
向下N LU a 供 DLUR 支V D > X Z c。

downstream_lu_detail.det_data.lu_sscp_stats.rcv_ru_size
最大 S U RU 大小。如果 **downstream_lu_detail.det_data.dspu_services** h 置为 AP_PU_CONCENTRATION，则 # t C字段。

downstream_lu_detail.det_data.lu_sscp_stats.send_ru_size
最大发M RU 大小。如果 **downstream_lu_detail.det_data.dspu_services** h 置为 AP_PU_CONCENTRATION，则 # t C字段。

downstream_lu_detail.det_data.lu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

downstream_lu_detail.det_data.lu_sscp_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

downstream_lu_detail.det_data.lu_sscp_stats.max_send_pac_win
< 终 + C字段 h 置为 c。

downstream_lu_detail.det_data.lu_sscp_stats.cur_send_pac_win
< 终 + C字段 h 置为 c。

downstream_lu_detail.det_data.lu_sscp_stats.max_rcv_pac_win
< 终 + C字段 h 置为 c。

downstream_lu_detail.det_data.lu_sscp_stats.cur_rcv_pac_win
< 终 + C字段 h 置为 c。

downstream_lu_detail.det_data.lu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ?。

downstream_lu_detail.det_data.lu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

downstream_lu_detail.det_data.lu_sscp_stats.send_data_bytes
发M D } # }] w D字Z }。

downstream_lu_detail.det_data.lu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

downstream_lu_detail.det_data.lu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

downstream_lu_detail.det_data.lu_sscp_stats.rcv_data_bytes
S U D } # }] w D字 Z } 。

downstream_lu_detail.det_data.lu_sscp_stats.sidh
会话 ID _ 字 Z 。

downstream_lu_detail.det_data.lu_sscp_stats.sidl
会话 ID M字 Z 。

downstream_lu_detail.det_data.lu_sscp_stats.odai
起 < 目 D X 址指 > 符。 * < 会话 1 , 如果 > X Z c | 含主 4 7 > , 则 BIND D 发M方 + C 字段 h 置为 c ; 而如果 BIND 发M方 G | 含次级 4 7 > D Z c , 则 Q C 字段 h 置为 1。

downstream_lu_detail.det_data.lu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 | G 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。 y P 8 v 字 Z 都 G P 效 D 。

downstream_lu_detail.det_data.ds_plu_stats.rcv_ru_size
最大 S U RU 大小。

downstream_lu_detail.det_data.ds_plu_stats.send_ru_size
最大发M RU 大小。

downstream_lu_detail.det_data.ds_plu_stats.max_send_btu_size
能发M D 最大 BTU 大小。

downstream_lu_detail.det_data.ds_plu_stats.max_rcv_btu_size
能 S U D 最大 BTU 大小。

downstream_lu_detail.det_data.ds_plu_stats.max_send_pac_win
会话中发M w = 窗 Z D 最大 _ 寸。

downstream_lu_detail.det_data.ds_plu_stats.cur_send_pac_win
会话中发M w = 窗 Z D 1 0 大小。

downstream_lu_detail.det_data.ds_plu_stats.max_rcv_pac_win
会话中 S U w = 窗 Z D 最大 _ 寸。

downstream_lu_detail.det_data.ds_plu_stats.cur_rcv_pac_win
会话中 S U w = 窗 Z D 1 0 大小。

downstream_lu_detail.det_data.ds_plu_stats.send_data_frames
发M D } # w }] 帧 D } ? 。

downstream_lu_detail.det_data.ds_plu_stats.send_fmd_data_frames
发M D } # w FMD }] 帧 D } ? 。

downstream_lu_detail.det_data.ds_plu_stats.send_data_bytes
发M D } # }] w D字 Z } 。

downstream_lu_detail.det_data.ds_plu_stats.rcv_data_frames
S U D } # w }] 帧 D } ? 。

downstream_lu_detail.det_data.ds_plu_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧 D } ? 。

downstream_lu_detail.det_data.ds_plu_stats.rcv_data_bytes
S U D } # }] w D字 Z } 。

downstream_lu_detail.det_data.ds_plu_stats.sidh
会话 ID _字Z。

downstream_lu_detail.det_data.ds_plu_stats.sidl
会话 ID M字Z。

downstream_lu_detail.det_data.ds_plu_stats.odai
起<目DX址指>符。1 ("会话1，如果>XZc |含主477>，则 BIND D发M方置C字段为c；而如果 BIND D发M方G |含次级477>DZc，则置C字段为 1。

downstream_lu_detail.det_data.ds_plu_stats.ls_name
同统计信息相关*D477>{F。 |G在>XI显>字符集中D一v 8 字Z字符串。 yP 8 v字Z都GP效D。

downstream_lu_detail.det_data.plu_stats.pacing_type
在下N PLU-SLU 会话中9CDSUw=类型。CN}能取 AP_NONE 或 AP_PACING_FIXED 值。

downstream_lu_detail.det_data.lu_sscp_pacing_type
LU-SSCP 会话中9CDSUw=。 |取值 AP_NONE。

downstream_lu_detail.det_data.us_plu_stats.send_ru_size
最大发M RU 大小。

downstream_lu_detail.det_data.us_plu_stats.max_send_btu_size
能发MD最大 BTU 大小。

downstream_lu_detail.det_data.us_plu_stats.max_rcv_btu_size
能SUD最大 BTU 大小。

downstream_lu_detail.det_data.us_plu_stats.max_send_pac_win
会话中发Mw=窗ZD最大_寸。

downstream_lu_detail.det_data.us_plu_stats.cur_send_pac_win
会话中发Mw=窗ZD10大小。

downstream_lu_detail.det_data.us_plu_stats.max_rcv_pac_win
会话中SUw=窗ZD最大_寸。

downstream_lu_detail.det_data.us_plu_stats.cur_rcv_pac_win
会话中SUw=窗ZD10大小。

downstream_lu_detail.det_data.us_plu_stats.send_data_frames
发MD}#w}]帧D}?。

downstream_lu_detail.det_data.us_plu_stats.send_fmd_data_frames
发MD}#w FMD }]帧D}?。

downstream_lu_detail.det_data.us_plu_stats.send_data_bytes
发MD}#}]wD字Z}。

downstream_lu_detail.det_data.us_plu_stats.rcv_data_frames
SUD}#w}]帧D}?。

downstream_lu_detail.det_data.us_plu_stats.rcv_fmd_data_frames
SUD}#w FMD }]帧D}?。

downstream_lu_detail.det_data.us_plu_stats.rcv_data_bytes
SUD}#}]wD字Z}。

downstream_lu_detail.det_data.us_plu_stats.sidh
会话 ID _字Z。如果 **downstream_lu_detail.det_data_.dspu_services** h置为 AP_PU_CONCENTRATION，则# t C字段。

downstream_lu_detail.det_data.us_plu_stats.sidl
会话 ID M字Z。如果 **downstream_lu_detail.det_data_.dspu_services** h置为 AP_PU_CONCENTRATION，则# t C字段。

downstream_lu_detail.det_data.us_plu_stats.odai
起<目D X址指>符。1 ("会话1，如果> X Z c | 含主4 7 >，则 BIND D发M方h置C字段为c；而如果 BIND D发M方G | 含次级4 7 > D Z c，则h置C字段为 1。如果 **downstream_lu_detail.det_data_.dspu_services** h置为 AP_PU_CONCENTRATION，则# t C字段。

downstream_lu_detail.det_data.us_plu_stats.ls_name
同统计信息相关* D 4 7 > { F。 | G在> X I 显>字符集中D一v 8 字Z字符串。 y P 8 v字Z都G P效D。如果 **downstream_lu_detail.det_data_dspu_services** h置为 AP_PU_CONCENTRATION，则# t C字段。

downstream_lu_detail.det_data.us_plu_stats.pacing_type
PLU-SLU 会话中9 C D S U w =类型。C N }能取 AP_NONE 或 AP_PACING_FIXED 值。

downstream_lu_detail.det_data.host_lu_name
下N LU 3 d = D主机 LU D { F，或G在O一次 PLU-SLU 会话活动1下N LU 3 d = D主机 LU D { F。 | I能k **def_data.host_lu_name** ；同，因为 def_data.host_lu_name I能G主机 LU X D { F。

downstream_lu_detail.det_data.host_pu_name
下N PU 3 d = D主机 PU D { F，或G在O一次 PLU-SLU 会话活动1下N PU 3 d = D主机 PU D { F。

downstream_lu_detail.def_data.description
资源5 w（如在 DEFINE_DOWNSTREAM_LU 或 DEFINE_DOWNSTREAM_LU_RANGE 中5 w D）。

downstream_lu_detail.def_data.nau_address
LU D网g I寻址%元X址，范围在 1--255 内。

downstream_lu_detail.def_data.dspu_name
k LU关* D PU D { F。b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。

downstream_lu_detail.def_data.host_lu_name
下N LU 3 d = D主机 LU 或主机 LU X D { F。M LU 而言，b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。M LU X而言，v人通信或通信服务器；为C字段指定字符集。C字段为J C DLUR D下N LU y # t。

downstream_lu_detail.def_data.allow_timeout
指定如果会话；活动D 1间达=在主机 LU 定义中指定D**超时**值，G否允许v人通信或通信服务器对C下N LU y 9 C D主机 LU ，1（AP_YES 或 AP_NO）。

downstream_lu_detail.def_data.delayed_logon

指定 v 人通信或通信服务器G否&延Y下N LU k主机 LU D，S，直至S U = Z 一 v 来自下N LU D }]。相反， + 发M一 v 模拟注 a 屏幕 = 下N LU（AP_YES 或 AP_NO）。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_INVALID_LIST_OPTION

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DOWNSTREAM_PU

C 动词 v J C Z 通信服务器。

QUERY_DOWNSTREAM_PU 返回 P 关下 N PU（C DEFINE_LS 动词定义 D）D 信息。

b 些信息以 * 要 q = 或详细信息 q = D P m 返回。如需 P 关某 v X 定 > X PU D 信息，或要获 C 几 v 段中 D P m 信息，& h 置 **dspu_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C 字段。

C PU P m I y] > X Z c 向下 N PU a 供 D 服务类型来过 K。要 b 样做，& + **dspu_services** 字段 h 置为 AP_PU_CONCENTRATION 或 AP_DLUR。

VCB a 构

```
typedef struct query_downstream_pu
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   attributes;        /* Verb attributes               */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   *buf_ptr;          /* pointer to buffer             */
        unsigned long   buf_size;          /* buffer size                   */
        unsigned long   total_buf_size;    /* total buffer size required    */
        unsigned short  num_entries;       /* number of entries             */
        unsigned short  total_num_entries; /* total number of entries       */
        unsigned char   list_options;      /* listing options               */
        unsigned char   reserv3;           /* reserved                      */
        unsigned char   dspu_name[8];      /* Downstream PU name            */
        unsigned char   dspu_services;     /* filter on DSPU services type  */
} QUERY_DOWNSTREAM_PU;

typedef struct downstream_pu_data
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   dspu_name[8];      /* PU name                       */
        unsigned char   description[RD_LEN];
                                           /* resource description          */
        unsigned char   ls_name[8];        /* Link name                     */
        unsigned char   pu_sscp_sess_active;
                                           /* Is PU-SSCP session active     */
        unsigned char   dspu_services;     /* DSPU service type             */
        SESSION _STATS  pu_sscp_stats;     /* SSCP-PU session stats         */
        unsigned char   reserva[20];       /* reserved                      */
} DOWNSTREAM_PU_DATA

typedef struct session_stats
{
        unsigned short  rcv_ru_size;       /* session receive RU size       */
        unsigned short  send_ru_size;      /* session send RU size          */
        unsigned short  max_send_btu_size; /* max send BTU size             */
        unsigned short  max_rcv_btu_size;  /* max rcv BTU size              */
        unsigned short  max_send_pac_win;  /* max send pacing win size      */
        unsigned short  cur_send_pac_win;  /* current send pacing win size  */
        unsigned short  max_rcv_pac_win;   /* max receive pacing win size   */
        unsigned short  cur_rcv_pac_win;   /* current receive pacing        */
                                           /* window size                   */
        unsigned long   send_data_frames;  /* number of data frames sent    */
        unsigned long   send_fmd_data_frames;
                                           /* num of FMD data frames sent   */
```

```
        unsigned long   send_data_bytes;   /* number of data bytes sent     */
        unsigned long   rcv_data_frames;   /* num data frames received      */
        unsigned long   rcv_fmd_data_frames;
                                           /* num of FMD data frames recvd  */
        unsigned long   rcv_data_bytes;    /* number of data bytes received */
        unsigned char   sidh;              /* session ID high byte          */
        unsigned char   sidl;              /* session ID low byte           */
        unsigned char   odai;              /* ODAI bit set                  */
        unsigned char   ls_name[8];        /* Link station name             */
        unsigned char   pacing_type;       /* type of pacing in use         */
} SESSION_STATS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_DOWNSTREAM_PU

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。

buf_size

y a供D缓e x D大小。返回D }] + ；，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。c值m > 无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回*要信息。

AP_DETAIL

返回详细信息。

指定D **dslu_name**（见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

QUERY_DOWNSTREAM_PU

dspu_name

} i 询D下N PU D { F。 b G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v字母 * 头）， " 以 EBCDIC U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

dspu_services

DSPU 服务过K器。如果h置为 AP_PU_CONCENTRATION，则v返回 PU 集中服务D下N LU。如果h置为 AP_DLUR，则v返回 DLUR 支V D LU。否则，如果h置为 AP_NONE，则返回y P下N LU D信息。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size

返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总 }。C值I以大Z **num_entries**。

downstream_pu_data.overlay_size

C项中D字Z }，即 = 下一v返回项（如果P D话）D偏移?。

downstream_pu_data.dspu_name

下N PU D { F。 b G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v字母 * 头）， " 以 EBCDIC U q R n d。

downstream_pu_data.description

资源5 w（如在 DEFINE_LS 中5 w D）。

downstream_pu_data.ls_name

4 7 > D { 字。 | G在 > X I 显 > 字符集中D一v 8 字Z字符串。y P 8 v字Z都G P效D。

downstream_pu_data.pu_sscp_sess_active

指 > = 下N PU D PU_SSCP 会话G否G活动D。要4置为 AP_YES，要4置为 AP_NO。

downstream_pu_data.dspu_services

指定 > X Z c a供x下N PU g越4 7 D服务。 | h置为下P值之一:

AP_PU_CONCENTRATION

向下N LU a供 PU 集中D > X Z c。

AP_DLUR

向下N LU a供 DLUR 支V D > X Z c。

downstream_pu_data.pu_sscp_stats.rcv_ru_size
最大S U RU 大小。如果 **downstream_lu_detail.det_data.dspu_services** h 置为 AP_PU_CONCENTRATION，则# t C字段。

downstream_pu_data.pu_sscp_stats.send_ru_size
最大发M RU 大小。如果 **downstream_lu_detail.det_data.dspu_services** h 置为 AP_PU_CONCENTRATION，则# t C字段。

downstream_pu_data.pu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

downstream_pu_data.pu_sscp_stats.max_rcv_btu_size
能S U D最大 BTU 大小。

downstream_pu_data.pu_sscp_stats.max_send_pac_win
<终+C字段h置为c。

downstream_pu_data.pu_sscp_stats.cur_send_pac_win
<终+C字段h置为c。

downstream_pu_data.pu_sscp_stats.max_rcv_pac_win
<终+C字段h置为c。

downstream_pu_data.pu_sscp_stats.cur_rcv_pac_win
<终+C字段h置为c。

downstream_pu_data.pu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ?。

downstream_pu_data.pu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

downstream_pu_data.pu_sscp_stats.send_data_bytes
发M D } # }] w D字Z }。

downstream_pu_data.pu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

downstream_pu_data.pu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

downstream_pu_data.pu_sscp_stats.rcv_data_bytes
S U D } # }] w D字Z }。

downstream_pu_data.pu_sscp_stats.sidh
会话 ID _字Z。

downstream_pu_data.pu_sscp_stats.sidl
会话 ID M字Z。

downstream_pu_data.pu_sscp_stats.odai
起<目D X址指>符。1 ("会话1，如果>X Z c |含主4 7 >，则 BIND D发M方置C字段为c；而如果 BIND D发M方G |含次级4 7 > D Z c，则置C字段为 1。

downstream_pu_data.pu_sscp_stats.ls_name
同统计信息相关* D 4 7 > { F。 |G在>X I 显>字符集中D一v 8 字Z字符串。 y P 8 v字Z都G P效D。

downstream_pu_data.pu_sscp_stats.pacing_type
在O N PU-SSCP 会话中 y 9 C D S U w = 类型。 C N } 能取 AP_NONE 值。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_PU_NAME

AP_INVALID_PU_TYPE
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_DSPU_TEMPLATE

C动词v J C Z通信服务器。

QUERY_DSPU_TEMPLATE 返回P关C Z隐= 4 7 O PU 集中D已定义D下N PU 模e D信息。C信息以P m返回。如需P关某v X定下N PU 模e D信息，或要获C几v段中D P m信息，& h置 **template_name** 字段。否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽T C字段。k N阅Z 12页D『i询Z c』，以获取关Z如何9 C P m q = D 3 0知6。

VCB a构

```
typedef struct query_dspu_template
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   attributes;        /* Verb attributes               */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   *buf_ptr;          /* pointer to buffer             */
        unsigned long   buf_size;          /* buffer size                   */
        unsigned long   total_buf_size;    /* total buffer size required    */
        unsigned short  num_entries;       /* number of entries             */
        unsigned short  total_num_entries; /* total number of entries       */
        unsigned char   list_options;      /* listing options               */
        unsigned char   reserv3;           /* reserved                      */
        unsigned char   template_name[8];  /* name of DSPU template         */
} QUERY_DSPU_TEMPLATE;

typedef struct dspu_template_data
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   template_name[8];  /* name of DSPU template         */
        unsigned char   description;       /* resource description          */
        unsigned char   reserv1[12];       /* reserved                      */
        unsigned short  max_instance;      /* max active template instances */
        unsigned short  active instance;   /* current active instances      */
        unsigned short  num_of_dslu_templates;
                                           /* number of DSLU templates      */
} DSPU_TEMPLATE_DATA;
```

？v **dspu_template_data** 后z **num_of_dslu_templates** 下N LU 模e。？v下N LU 模e D q = 如下。

```
typedef struct dslu_template_data
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   reserv1[2];        /* reserved                      */
        DSLU_TEMPLATE   dslu_template;     /* downstream LU template        */
} DSLU_TEMPLATE_DATA;

typedef struct dslu_template
{
        unsigned char   min_nau;           /* min NAU address in range      */
        unsigned char   max_nau;           /* max NAU address in range      */
        unsigned char   reserv1[10];       /* reserved                      */
        unsigned char   host_lu[8];        /* host LU or pool name          */
} DSLU_TEMPLATE;
```

QUERY_DSPU_TEMPLATE

提供N数

＆C L序a供下P N }:

opcode

AP_QUERY_DSPU_TEMPLATE

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。＆C L序I以+ }] =加在 VCB D末尾，在b种i v下，**buf_ptr** X须h置为 NULL。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m >无限制。

list_options

m w在P m信息中＆C返回2 4:

指定D **template_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

template_name

DSPU 模e D { F。|G一v以> X I显>字符集m > D 8 字Z字符串。如果**list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size
返回值m w返回 y P ； k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大 Z **buf_size**。

num_entries
5 际返回 D 项 D } ？。

total_num_entries
已 - 返回 D 项 D 总 } 。 C 值 I 以大 Z **num_entries**。

dspu_template_data.overlay_size
C 项中 D 字 Z } （ | 括任何下 N LU 模 e , 即 = 下一 v 返回项（如果 P D 话） D 偏移？)。

dspu_template_data.template_name
DSPU 模 e D { F 。 | G 一 v 以 > X I 显 > 字符集 m > D 8 字 Z 字符串。

dspu_template_data.description
资源 5 w（如在 QUERY_DSPU_TEMPLATE 中 5 w D)。

dspu_template_data.max_instance
b G I 以同 1 为活动 D 模 e D 最大 5 例 } 。

dspu_template_data.active_instance
b G 1 0 活动模 e D 5 例 } 。

dspu_template_data.num_of_dslu_templates
C Z C 下 N PU 模 e D 下 N LU 模 e D } ？。 t z 在 C 字段后 f D G **num_of_dslu_templates_application_id** 项, 注 a 为 9 c 类 p D ？ v & C L 序都 P 一 v b 样 D 项。

dslu_template_data.overlay_size
C 项中 D 字 Z } （即 = 下一 v 返回项（如果 P D 话）D 偏移？)。

dslu_template_data.dslu_template.min_nau
范围内 D 最小 NAU X 址。

dslu_template_data.dslu_template.max_nau
范围内 D 最大 NAU X 址。

def_data.allow_timeout
指定如果会话；活动 D 1 间达 = 在主机 LU 定义中指定 D **超时**值, G 否允许 L 序对 C 下 N LU y 9 C D 主机 LU , 1（AP_YES 或 AP_NO)。

def_data.delayed_logon
指定 L 序 G 否 & 延 Y 下 N LU k 主机 LU D , S , 直至 S U = Z 一 v 来自下 N LU D }] 。相反, 发 M 一 v 模拟注 a 屏幕 = 下 N LU（AP_YES 或 AP_NO)。

dslu_template_data.dslu_template.host_lu_name
一定范围内 D y P 下 N LU + 3 d = D 主机 LU 或主机 LU X D { F 。 b G v 8 字 Z D 字母 } 字集 D A-EBCDIC 型 EBCDIC 字符串（以字母 * <), I EBCDIC U q R n d 。

如果因 N } 错误而 9 动词；能执行, 则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TEMPLATE_NAME

AP_INVALID_LIST_OPTION

如果因相关 START_NODE N } 未 h 置而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_FOCAL_POINT

QUERY_FOCAL_POINT 返回 P 关为 v 人通信或通信服务器 y 知 D 9 c D信息。

C信息以 P m返回。如需 P 关某 v X定 9 c 类 p D信息，或要获 C 几 v 『段』中 D P m信息，& h 置 **ms_category** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C字段。 k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

"： 如果；P R = 9 c，则 + 返回一 v FP_DATA a 构，R 其中 D **fp_data.fp_type** h 置为 AP_NO_FP。见下 P a 构。

VCB a 构

```
typedef struct query_focal_point
{
        unsigned short  opcode;            /* verb operation code          */
        unsigned char   reserv2;           /* reserved                     */
        unsigned char   format;            /* format                       */
        unsigned short  primary_rc;        /* primary return code          */
        unsigned long   secondary_rc;      /* secondary return code        */
        unsigned char   *buf_ptr;          /* pointer to buffer            */
        unsigned long   buf_size;          /* buffer size                  */
        unsigned long   total_buf_size;    /* total buffer size required   */
        unsigned short  num_entries;       /* number of entries            */
        unsigned short  total_num_entries; /* total number of entries      */
        unsigned char   list_options;      /* listing options              */
        unsigned char   reserv3;           /* reserved                     */
        unsigned char   ms_category[8];    /* name of MS category          */
} QUERY_FOCAL_POINT;

typedef struct fp_data
{
        unsigned short  overlay_size;      /* size of this entry           */
        unsigned char   ms_appl_name[8];   /* focal point application name */
        unsigned char   ms_category[8];    /* focal point category         */
        unsigned char   description[RD_LEN];
                                           /* resource description         */
        unsigned char   fp_fqcp_name[17];  /* focal pt fully qual CP name  */
        unsigned char   bkup_appl_name[8]; /* backup focal pt appl name    */
        unsigned char   bkup_fp_fqcp_name[17];
                                           /* backup FP fully qualified    */
                                           /* CP name                      */
        unsigned char   implicit_appl_name[8];
                                           /* implicit FP appl name        */
        unsigned char   implicit_fp_fqcp_name[17];
                                           /* implicit FP fully            */
                                           /* qualified CP name            */
        unsigned char   fp_type;           /* focal point type             */
        unsigned char   fp_status;         /* focal point status           */
        unsigned char   fp_routing;        /* type of MDS routing to use   */
        unsigned char   reserva[20];       /* reserved                     */
        unsigned short  number_of_appls;   /* number of applications       */
} FP_DATA;
```

？v **fp_data** 后 z **number_of_appls** & C L 序 { F 。？v & C L 序 { F D q = 如下:

```
typedef struct application_id
{
        unsigned char    appl_name[8];     /* application name             */
} APPLICATION_ID;
```

QUERY_FOCAL_POINT

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_FOCAL_POINT

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m > 无限制。

list_options

| m w P m信息中&返回2 4: 指定D **ms_category**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

ms_category

管理服务类p。|要4 G如 SNA 管理服务中h v D管理服务类p D a构定义D 4 字Z值（以 EBCDIC U q R n d）之一，要4 G一v 8 字Z 1134 型EBCDIC 2装定义D{ F。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size

返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总}。C值I以大Z **num_entries**。

fp_data.overlay_size

C项中D字Z }（ | 括任何&C L序{ F，即 = 下一 v 返回项（如果 P D话）D偏移？）。

fp_data.ms_appl_name

１０活动D９ｃ&C L序D{ F。 | 要４G如 SNA 管理服务中ｈｖD管理服务&C L序D a 构定义D 4 字Z值（以 EBCDIC U q R n d）之一，要４G一ｖ 8 字Z 1134 型 EBCDIC 2装定义D{ F。

fp_data.ms_category

管理服务类p。 | 要４G如 SNA 管理服务中ｈｖD管理服务类p D a 构定义D 4 字Z值（以 EBCDIC U q R n d）之一，要４G一ｖ 8 字Z 1134 型 EBCDIC 2装定义D{ F。

fp_data.description

资源５w（如在 DEFINE_FOCAL_POINT 中５wD）。 | G一v以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

fp_data.fp_fqcp_name

１０活动９ｃD全限定X制ｃ{ F。{ F $度为 17 v字Z，"以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。（？v{ F D $度最多I为 8 v字Z，"；含６入U q。）

fp_data.bkup_appl_name

８份９ｃ&C L序D{ F。 | 要４G如 SNA 管理服务中ｈｖD管理服务&C L序D a 构定义D 4 字Z值（以 EBCDIC U q R n d）之一，要４G一ｖ 8 字Z 1134 型 EBCDIC 2装定义D{ F。

fp_data.bkup_fp_fqcp_name

８份９ｃD全限定X制ｃ{ F。{ F $度为 17 v字Z，"以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。（？v{ F D $度最多I为 8 v字Z，"；含６入U q。）

fp_data.implicit_appl_name

隐 = ９ｃ&C L序D{ F（C DEFINE_FOCAL_POINT 动词指定D）。 | 要４G如 SNA 管理服务中ｈｖD管理服务&C L序D a 构定义D 4 字Z值（以 EBCDIC U q R n d）之一，要４G一ｖ 8 字Z 1134 型 EBCDIC 2装定义D{ F。如果隐 = ９ｃG１０活动D９ｃ，则C字段ｋ **ms_appl_name** 相同。

fp_data.bkup_fp_fqcp_name

隐 = ９ｃD全限定X制ｃ{ F（C DEFINE_FOCAL_POINT 动词指定D）。{ F $度为 17 v字Z，"以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。（？v{ F D $度最多I为 8 v字Z，"；含６入U q。）如果隐 = ９ｃG１０活动D９ｃ，则C字段ｋ **fp_fqcp_name** 相同。

fp_data.fp_type

９ｃD类型。N < *SNA 管理服务*，以获C | 详细资O。 | I h置为下P值之一：

AP_EXPLICIT_PRIMARY_FP
AP_BACKUP_FP
AP_DEFAULT_PRIMARY_FP
AP_IMPLICIT_PRIMARY_FP

AP_DOMAIN_FP
AP_HOST_FP
AP_NO_FP

fp_data.fp_status
9 c D状，。 | I h置为下P值之一:

AP_NOT_ACTIVE
C 9 c 1 0 ；活动。

AP_ACTIVE
C 9 c 1 0活动。

AP_PENDING
C 9 c G暂挂活动D。 b种 i v发 z 在 + 隐 = k s 发M x 9 c 之后，S U = 响&之 0 。

AP_NEVER_ACTIVE
！管已 S \ K C类 p D & C L 序注 a， + G对 Z 指定类 p 仍无 9 c 信息 I C。

fp_data.fp_routing
7 I 选择D类型，在 9 C MDS 传M + }] 发M x 9 c 1， &C L 序& C指定C值。

AP_DEFAULT
缺！ 7 I 选择C Z 传M MDS_MU 至 9 c 。

AP_DIRECT
MDS_MU + 在会话O直 S 传] = 9 c 。

fp_data.number_of_appls
注 a C 9 c 类 p D & C L 序D } ? 。 t z 在C字段后D G **number_of_appls application_id** 项， b些项G ? v注 a 9 c 类 p D & C L 序D项。

appl_name
注 a 9 c 类 p D & C L 序D { F 。 | 要 4 G如 SNA 管理服务中 h v D管理服务 & C L 序D a 构定义D 4 字Z 值（以 EBCDIC U q R n d ）之一，要 4 G一 v 8 字Z 1134 型 EBCDIC 2 装定义D { F 。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_MS_CATEGORY

AP_INVALID_LIST_OPTION

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N } :

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_HPR_STATS

C动词ｖＪＣＺ通信服务器。

QUERY_HPR_STATS 返回ｈｖＺｃＤ HPR 性能D统计信息。 QUERY_HPT_STATS ｖＩ支V RTP ~ＤＺｃ支V。

VCB a构

```
typedef struct query_hpr_stats
{
        unsigned short  opcode;             /* verb operation code         */
        unsigned char   reserv2;            /* reserved                    */
        unsigned char   format;             /* format                      */
        unsigned short  primary_rc;         /* primary return code         */
        unsigned long   secondary_rc;       /* secondary return code       */
        unsigned COUNTER
                        num-orig_rs_sent;   /* RS requests sent as origin  */
        unsigned COUNTER
                        num_orig_rs_rej;    /* RS rejections at orign      */
        unsigned COUNTER
                        num_inter_rs_rcvd;  /* Intermediate RS requests    */
        unsigned COUNTER
                        num_inter_rs_rej;   /* Intermediate RS rejections  */
        unsigned COUNTER
                        num_dest_rs_rcvd;   /* RS reqs as destination      */
        unsigned COUNTER
                        num_dest_rs_rej;    /* RS rej sent as destination  */
        unsigned char   reserv[28];         /* reserved                    */
} QUERY_HPR_STATS;
```

提供N数

＆ＣＬ序a供下ＰＮ}:

opcode

AP_QUERY_HPR_STATS

format

ｊ６ VCB Ｄｑ＝。+C字段h置为c，以指定Ｏｆ ＰｖＤ VCB Ｄｆ＞。

返回N数

如果动词Ｉ功执行，则L序返回下ＰＮ}:

primary_rc

AP_OK

num_orig_rs_sent

自从Ｚｃ启动后， ｚｚＺＣＺｃＤ HPR ７Ｉｈ置ｋｓ发MD总}。

num_orig_rs_rej

自从Ｚｃ启动后，在ＣＺｃｚｚＤ、"；其|Ｚｃ\ｘＤ HPR ７Ｉｈ置ｋｓＤ总}。

num_inter_rs_rcvd

QUERY_ISR_SESSION

C动词v J C Z通信服务器。

QUERY_ISR_SESSION v在网g Z c O 9 C， " R |返回关Z会话（网g Z c向其a供中间会话7 I选择）D P m信息。

b些信息以*要q =或详细信息q = D P m返回。如需关Z某v X定会话D信息，或_获取几v『段』中D P m信息， & C h置 **fqpcid** a构中D字段。否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST）， C a构中D字段+；忽T。 k N阅Z 12页D『i询Z c』，以获取关Z如何9 C P m q = D 3 0知6。

C P mW先I **fqpcid.pcid** 定序，然后4 **fqpcid.fqcp_name** x行 EBCDIC 词d `纂定序。 4 **fqpcid.pcid_name** 定序W先4 { F $度，然后对相同$度D { F x行 ASCII 词d `纂定序（ y] IBM D 6611 APPN MIB 定序）。如果选中K AP_LIST_FROM_NEXT，则返回D P m 4定义D次序（无[指定D项G否存在）从下一项*<。

fqpcid a构D q = G一v 8字Z D过L相关因子j 6（PCID）和会话发z器D网g全限定 CP { F。

}K？v会话D详细信息，如果在 START_NODE N }O指定C项，则返回7 I选择X制向？（RSVC）。 C RSVC 通过网g定义K会话以逐段形= x行7 I。

VCB a构

格式 2

```
typedef struct query_isr_session
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   *buf_ptr;          /* pointer to buffer            */
   unsigned long   buf_size;          /* buffer size                  */
   unsigned long   total_buf_size;    /* total buffer size required   */
   unsigned short  num_entries;       /* number of entries            */
   unsigned short  total_num_entries; /* total number of entries      */
   unsigned char   list_options;      /* listing options              */
   unsigned char   session_type;      /* is this query for DLUR or    */
                                      /* regular ISR sessions?        */
   FQPCID          fqpcid;            /* fully qualified procedure    */
                                      /* correlator ID                */
} QUERY_ISR_SESSION;

typedef struct isr_session_summary
{
   unsigned short  overlay_size;      /* size of this entry           */
   FQPCID          fqpcid;            /* fully qualified procedure    */
                                      /* correlator ID                */
} ISR_SESSION_SUMMARY;

typedef struct isr_session_detail
{
   unsigned short  overlay_size;      /* size of this entry           */
   FQPCID          fqpcid;            /* fully qualified procedure    */
```

```
   unsigned short  sub_overlay_size;  /* offset to appended RSCV      */
                                      /* correlator ID                */
   unsigned char   trans_pri;         /* Transmission priority:       */
   unsigned char   cos_name[8];       /* Class-of-service name        */
   unsigned char   ltd_res;           /* Session spans a limited      */
   unsigned char   reserv1[8];        /* reserved                    */
                                      /* resource                     */
   SESSION_STATS   pri_sess_stats;    /* primary hop session stats    */
   SESSION_STATS   sec_sess_stats;    /* secondary hop session        */
                                      /* statistics                   */
   unsigned char   sess_lu_type;      /* session LU type              */
   unsigned char   sess_lu_level;     /* session LU level             */
   unsigned char   pri_tg_number;     /* Primary session TG number    */
   unsigned char   sec_tg_number;     /* Secondary session TG number  */
   unsigned long   rtp_tcid;          /* RTP TC identifier            */
   unsigned long   time_active;       /* time elapsed since           */
                                      /* activation                   */
   unsigned char   isr_state;         /* current state of ISR session */
   unsigned char   reserv2[11];       /* reserved                     */
   unsigned char   mode_name[8];      /* mode name                    */
   unsigned char   pri_lu_name[17];   /* primary LU name              */
   unsigned char   sec_lu_name[17];   /* secondary LU name            */
   unsigned char   pri_adj_cp_name[17];
                                      /* primary stage adj CP name    */
   unsigned char   sec_adj_cp_name[17];
                                      /* secondary stage adj CP name  */
   unsigned char   reserv3[3];        /* reserved                     */
   unsigned char   rscv_len;          /* Length of following RSCV     */
} ISR_SESSION_DETAIL;

typedef struct fqpcid
{
   unsigned char   pcid[8];           /* pro correlator identifier    */
   unsigned char   fqcp_name[17];     /* orig's network qualified     */
                                      /* CP name                      */
   unsigned char   reserve3[3];       /* reserved                     */
} FQPCID;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;       /* session receive RU size      */
   unsigned short  send_ru_size;      /* session send RU size         */
   unsigned short  max_send_btu_size; /* Maximum send BTU size        */
   unsigned short  max_rcv_btu_size;  /* Maximum rcv BTU size         */
   unsigned short  max_send_pac_win;  /* Max send pacing window size  */
   unsigned short  cur_send_pac_win;  /* Curr send pacing window size */
   unsigned short  max_rcv_pac_win;   /* Max receive pacing win size  */
   unsigned short  cur_rcv_pac_win;   /* Curr rec pacing window size  */
   unsigned long   send_data_frames;  /* Number of data frames sent   */
   unsigned long   send_fmd_data_frames;
                                      /* num of FMD data frames sent  */
   unsigned long   send_data_bytes;   /* Number of data bytes sent    */
   unsigned long   rcv_data_frames;   /* Num data frames received     */
   unsigned long   rcv_fmd_data_frames;
                                      /* num of FMD data frames recvd */
   unsigned long   rcv_data_bytes;    /* Num data bytes received      */
   unsigned char   sidh;              /* Session ID high byte         */
   unsigned char   sidl;              /* Session ID low byte          */
   unsigned char   odai;              /* ODAI bit set                 */
   unsigned char   ls_name[8];        /* Link station name            */
   unsigned char   pacing_type;       /* type of pacing in use        */
} SESSION_STATS;
```

VCB a 构

格式 1（后备 6 别）

```
typedef struct isr_session_detail
{
   unsigned short  overlay_size;      /* size of this entry           */
   FQPCID          fqpcid;            /* fully qualified procedure    */
   unsigned short  sub_overlay_size;  /* offset to appended RSCV      */
                                      /* correlator ID                */
   unsigned char   trans_pri;         /* Transmission priority:       */
   unsigned char   cos_name[8];       /* Class-of-service name        */
   unsigned char   ltd_res;           /* Session spans a limited      */
   unsigned char   reserv1[2];        /* reserved                    */
                                      /* resource                     */
```

VCB a 构

格式 0（后备 6 别）

```
typedef struct isr_session_detail
{
   unsigned short  overlay_size;      /* size of this entry           */
   FQPCID          fqpcid;            /* fully qualified procedure    */
   unsigned char   trans_pri;         /* Transmission priority:       */
   unsigned char   cos_name[8];       /* Class-of-service name        */
   unsigned char   ltd_res;           /* Session spans a limited      */
   unsigned char   reserv1[8];        /* reserved                     */
                                      /* resource                     */
   SESSION_STATS   pri_sess_stats;    /* primary hop session stats    */
   SESSION_STATS   sec_sess_stats;    /* secondary hop session        */
                                      /* statistics                   */
   unsigned char   reserv3[3];        /* reserved                     */
   unsigned char   reserva[20];       /* reserved                     */
   unsigned char   rscv_len;          /* Length of following RSCV     */
} ISR_SESSION_DETAIL;
```

"： ISR 会话细 Z 2 G后 z I *SNA q* =定义 D 7 I 选择 X 制向？（RSCV）。C X 制向？通过网 g 定义 K 会话 7 I，"在 BIND O传 M。1 Z c 启动 1，v 定 G 否 | 含 C RSCV（作为 START_NODE D 一 v 选项），" R 以后 I 以 9 C DEFINE_ISR_STATS 来 P 换。如果已 9 C C 动词指定；要存储 RSCV，那 4 + **rscv_len** h 置为 c。

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_ISR_SESSION

format

j 6 VCB D q = 和返回 2 G D q =。+ 此字段 h 置为 c，以指定在 O f P v VCB 和 2 G D f >。

buf_ptr

指向 C Z 写入 P m 信息 D 缓 e x D 指 k。& C L 序 I + }] m 加 = VCB D 末尾（X 须 + **buf_ptr** h 置为 NULL）。

buf_size

y a 供 D 缓 e x D 大小。返回 D }] + ; ，过 b v 大小。

num_entries

要返回 D 项 D 最大 } ?。项 D } ? ；要，过 b v 值。c 值 m > 无限制。

list_options

m w 在 P m 信息中 & C 返回 2 4。

AP_SUMMARY

v 返回 * 要信息。

AP_DETAIL

返回详细信息。

指定 D **fqpcid**（见下 P N }）m > w 引值，C w 引值 C Z 指定要返回 D 5 际信息 D 起 < c。

AP_FIRST_IN_LIST
忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT
返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE
返回P m从w引值y指定D那项* <。

session_type
x行C动词i询 DLUR 维护会话，或#规 ISR 会话?

AP_ISR_SESSION ISR 会话
AP_DLUR_SESSIONS DLUR 会话

fqpcid.pcid
过L相关因子 ID。b G v 8 字Z D . y x制字符串。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

fqpcid.pcid_name
全限定X制c D { 字。C { 字为 17 字Z $ R I EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(? v { F D $ 度最多I为 8 v字Z，"；含6人U q。)如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总}。C值I以大Z **num_entries**。

isr_session_summary.overlay_size
C项中D字Z }，即=下一v返回项(如果P D话)D偏移?。

isr_session_summary.fqpcid.pcid
过L相关因子 ID。

isr_session_summary.fqpcid.fqcp_name
全限定X制c D { 字。C { 字为 17 字Z $ R I EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c，S。(? v { F D $ 度最多I为 8 v字Z，"；含6人U q。)

isr_session_detail.overlay_size

C项中D字Z }（ | 括任何 = 加 RSCV），即 = 下一 v 返回项（如果 P D话）D偏移？。

isr_session_detail.sub_overlay_size

C字段 x v K C细 Z 2 G D大小。如果 = 加K RSCV, 那 4 b G = RSCV D * 头D偏移？。C字段 I 以大 Z H Z 一 v 详细 a 构D q = D大小（允许以后扩 9 ）。

isr_session_detail.fqpcid.pcid

过L 相关因子 ID。

isr_session_detail.fqpcid.fqcp_name

全限定X制 c D { 字。C { 字为 17 字 Z $ R I EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（？v { F D $ 度最多 I 为 8 v字Z， " ；含 6 入U q。）

session_detail.trans_pri

传 d E 先级。 | h 置为下 P 值之一:

AP_LOW
AP_MEDIUM
AP_HIGH
AP_NETWORK

session_detail.cos_name

服务级 { F。b G一 v 8 字 Z 字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。

session_detail.ltd_res

指定会话G否 9 C \ 限制D资源 4 7（AP_YES 或 AP_NO）。

isr_session_detail.pri_sess_stats.rcv_ru_size

最大 S U RU 大小。

isr_session_detail.pri_sess_stats.send_ru_size

最大发M RU 大小。

isr_session_detail.pri_sess_stats.max_send_btu_size

I 在主会话中继段O发M D最大 BTU 大小。

isr_session_detail.pri_sess_stats.max_rcv_btu_size

I 在主会话中继段O S U D最大 BTU 大小。

isr_session_detail.pri_sess_stats.max_send_pac_win

主会话中继段O发M w = 窗 Z D最大 _ 寸。

isr_session_detail.pri_sess_stats.cur_send_pac_win

主会话中继段O发M w = 窗 Z D 1 0大小。

isr_session_detail.pri_sess_stats.max_rcv_pac_win

主会话中继段O S U w = 窗 Z D最大 _ 寸。

isr_session_detail.pri_sess_stats.cur_rcv_pac_win

主会话中继段O S U w = 窗 Z D 1 0大小。

isr_session_detail.pri_sess_stats.send_data_frames

主会话中继段O发M D } # w }] 帧D } ？。

isr_session_detail.pri_sess_stats.send_data_frames
主会话中继段O发MD } # w }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回= C字段中。

isr_session_detail.pri_sess_stats.send_fmd_data_frames
主会话中继段O发MD } # FMD }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回= C字段中。

isr_session_detail.pri_sess_stats.send_data_bytes
主会话中继段O发MD } # }] w D字Z }。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回= C字段中。

isr_session_detail.pri_sess_stats.rcv_data_frames
主会话中继段O S U D } # }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回= C字段中。

isr_session_detail.pri_sess_stats.rcv_fmd_data_frames
主会话中继段O S U D } # FMD }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回= C字段中。

isr_session_detail.pri_sess_stats.rcv_data_bytes
主会话中继段O S U D } # }] w D字Z }。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回= C字段中。

isr_session_detail.pri_sess_stats.sidh
会话 ID _字Z。

isr_session_detail.pri_sess_stats.sidl
会话 ID M字Z。

isr_session_detail.pri_sess_stats.odai
原<目D X址指>符。1启动会话1，如果> X Z c | 含主4 7 >，则 BIND D发M方h置C字段为c。如果 BIND D发M方G | 含从4 7 > D Z c，则h置C字段为 1。

isr_session_detail.pri_sess_stats.ls_name
同统计信息相关* D 4 7 > { F。 | G在> X I 显>字符集中D一v 8 字Z字符串。 y P D 8 v字Z G P效D。C字段能C Z +会话D统计信息k会话 }] w过D 4 7关* 起来。

isr_session_detail.sec_sess_stats.rcv_ru_size
最大S U RU 大小。

isr_session_detail.pri_sess_stats.pacing_type
主会话中9 C D S U w =类型。C N } I 以取 AP_NONE、AP_PACING_FIXED 或 AP_PACING_ADAPTIVE H值。

isr_session_detail.sec_sess_stats.send_ru_size
最大发M RU 大小。

isr_session_detail.sec_sess_stats.max_send_btu_size
I 在次级会话中继段O发MD最大 BTU 大小。

isr_session_detail.sec_sess_stats.max_rcv_btu_size
I 在次级会话中继段O S U D最大 BTU 大小。

isr_session_detail.sec_sess_stats.max_send_pac_win
次级会话中继段O发M w =窗Z D最大_寸。

isr_session_detail.sec_sess_stats.cur_send_pac_win
次级会话中继段O发Mw = 窗Z D 1 0大小。

isr_session_detail.sec_sess_stats.max_rcv_pac_win
次级会话中继段O S U w = 窗Z D最大 _ 寸。

isr_session_detail.sec_sess_stats.cur_rcv_pac_win
次级会话中继段O S U w = 窗Z D 1 0大小。

isr_session_detail.sec_sess_stats.send_data_frames
次级会话中继段O发MD } # }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回 = C字段中。

isr_session_detail.sec_sess_stats.send_fmd_data_frames
次级会话中继段O发MD } # FMD }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回 = C字段中。

isr_session_detail.sec_sess_stats.send_data_bytes
次级会话中继段O发MD } # }] wD字Z }。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回 = C字段中。

isr_session_detail.sec_sess_stats.rcv_data_frames
次级会话中继段O S U D } # }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回 = C字段中。

isr_session_detail.sec_sess_stats.rcv_fmd_data_frames
次级会话中继段O S U D } # FMD }] 帧D } ?。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回 = C字段中。

isr_session_detail.sec_sess_stats.rcv_data_bytes
次级会话中继段O S U D } # }] wD字Z }。如果9 C DEFINE_ISR_STATS 而9统计信息D集合；{ C，则+ c返回 = C字段中。

isr_session_detail.sec_sess_stats.sidh
会话 ID _字Z。

isr_session_detail.sec_sess_stats.sidl
会话 ID M字Z（来自 LFSID)。

isr_session_detail.sec_sess_stats.odai
原<目D X址指>符。1启动会话1，如果>X Z c |含主4 7 >，则 BIND D发M方h置C字段为c。如果 BIND D发M方G |含从4 7 > D Z c，则h置C字段为 1。

isr_session_detail.sec_sess_stats.ls_name
同统计信息相关* D 4 7 >{ F。 |G在>X I显>字符集中D一v 8 字Z字符串。y P 8 v字Z都G P效D。C字段能C Z +中间会话D统计信息k X b 4 7 >关*起来。

isr_session_detail.sec_sess_stats.pacing_type
主会话中9 C D S U w = 类型。C N } I以取 AP_NONE、AP_PACING_FIXED 或 AP_PACING_ADAPTIVE H值。

isr_session.detail.sess_lu_type
在 BIND O指定D会话D LU 类型。C字段取下P其中一v值:

AP_LU_TYPE_0
AP_LU_TYPE_1
AP_LU_TYPE_2
AP_LU_TYPE_3
AP_LU_TYPE_4
AP_LU_TYPE_6
AP_LU_TYPE_7
AP_LU_TYPE_UNKNOWN
（LU 类型 5 ；P意！T。）

+ <终返回 AP_LU_TYPE_UNKNOWN, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.sess_lu_level

会话D LU 级p。 C字段取下P其中一v值:

AP_LU_LEVEL_0
AP_LU_LEVEL_1
AP_LU_LEVEL_2
AP_LU_LEVEL_UNKNOWN

如果 LU 类型；G 6, 则+C字段h置为 AP_LU_LEVEL_0。 + <终返回 AP_LU_LEVEL_UNKNOWN, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.pri_tg_number

k 4 7相关*D TG 号I主会话中继段i历。如果主会话W段i历 RTP ，S, 则返回c。 + <终返回c, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.sec_tg_number

k 4 7相关*D TG 号I主会话中继段i历。如果主会话W段i历 RTP ，S, 则返回c。 + <终返回c, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.rtp_tcid

RTP ，S D > X TC ID, 1 C ISR 会话组I ANR/ISR _ g D ?件1返回。在其 | i v下, + C字段h置为c。 + <终返回c, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.time_active

自从会话激活后D - 过1间, 以Y分之一k b ?。 + <终返回c, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.isr_state

会话D 1 0状，。 C字段h置为下P值之一:

AP_ISR_INACTIVE
AP_ISR_PENDING_ACTIVE
AP_ISR_ACTIVE
AP_ISR_PENDING_INACTIVE

isr_session.detail.mode_name

会话D方={。 b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）, "以 EBCDIC U q R n d。 + <终返回全二x制c, }非9 C DEFINE_ISR_STATS 启C { F D集合。

isr_session.detail.pri_lu_name

会话D主 LU ｛ F。｛ F $度为 17 v字Z， "以 EBCDIC U q R n d。C ｛字G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I。？v ｛ F最多I为 8 v字Z， "；含6入U q。如果b v ｛ F ； I C，则全二x制c返回 = C字段中。 + <终返回全二x制c， ｝非9 C DEFINE_ISR_STATS 启C ｛ F D集合。

isr_session.detail.sec_lu_name

会话D次级 LU ｛ F。｛ F $度为 17 v字Z， "以 EBCDIC U q R n d。C ｛字G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I。？v ｛ F最多I为 8 v字Z， "；含6入U q。如果b v ｛ F ； I C，则全二x制c返回 = C字段中。 + <终返回全二x制c， ｝非9 C DEFINE_ISR_STATS 启C ｛ F D集合。

isr_session.detail.pri_adj_cp_name

C会话D主W段Z | CP ｛ F。如果主会话W段i历 RTP ，S，则返回远L RTP 端c D CP ｛ F。｛ F $度为 17 v字Z， "以 EBCDIC U q R n d。C ｛字G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I。？v ｛ F最多I为 8 v字Z， "；含6入U q。如果b v ｛ F ； I C，则全二x制c返回 = C字段中。 + <终返回全二x制c， ｝非9 C DEFINE_ISR_STATS 启C ｛ F D集合。

isr_session.detail.sec_adj_cp_name

C会话D次级W段Z | CP ｛ F。如果次级会话W段i历 RTP ，S，则返回远L RTP 端c D CP ｛ F。｛ F $度为 17 v字Z， "以 EBCDIC U q R n d。C ｛字G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I。？v ｛ F最多I为 8 v字Z， "；含6入U q。如果b v ｛ F ； I C，则全二x制c返回 = C字段中。 + <终返回全二x制c， ｝非9 C DEFINE_ISR_STATS 启C ｛ F D集合。

如果因N ｝错误而9动词；能执行，则L序返回下P N ｝:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_FQPCID

AP_INVALID_LIST_OPTION
AP_INVALID_SESSION_TYPE

如果因相关 START_NODE N ｝未h置而9动词；能执行，则L序返回下P N ｝:

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果因Z c ； P ("（ _ P网g Z c支V ）而9动词；能执行，则L序返回下P N ｝:

primary_rc

AP_INVALID_VERB

如果因Z c P未启动而9动词；能执行，则L序返回下P N ｝:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_LOCAL_LU

QUERY_LOCAL_LU 返回关Z > X LU D信息。 I 以发v QUERY_LOCAL_LU 以检w关Z v人通信或通信服务器X制c LU D信息。

b些信息以 * 要q = 或详细信息q = D P m返回。如需关Z某v X定 > X LU D信息，或 _ 获取几v『段』中D P m信息， & C h置 **lu_name** 或 **lu_alias** 字段。如果 **lu_name** 字段非c，则 | + C Z确定w引。如果 **lu_name** 字段h置为全c，**lu_alias** + C Z确定w引。如果 **lu_name** 和 **lu_alias** 字段都h置为全c，则 + 9 C k X制c（缺！ LU）相关 * D LU。如果 **list_options** 字段h置为 AP_FIRST_IN_LIST，那4忽T b = v字段。（在b种i v下，如果h置K AP_LIST_BY_ALIAS **list_options**，则4 LU p { 排序，否则， 4 LU { F 排序）。 k N阅Z12页D『 i 询Z c 』，以获取关Z如何9 C P m q = D 3 0知6。

C P m y] 指定D选项4 **lu_alias** 或 **lu_name** 排序。字段4 U EBCDIC 词d ` 纂定序x行排序。

要返回D > X LU D P m I 以y] k | G相关 * D PU { F x行8选。在b种i v下，& h置 **pu_name** 字段（否则， C字段& h置为全c）。

VCB a构

格式 1

```
typedef struct query_local_lu
{
        unsigned short  opcode;            /* verb operation code              */
        unsigned char   reserv2;           /* reserved                         */
        unsigned char   format;            /* format                           */
        unsigned short  primary_rc;        /* primary return code              */
        unsigned long   secondary_rc;      /* secondary return code            */
        unsigned char   *buf_ptr;          /* pointer to buffer                */
        unsigned long   buf_size;          /* buffer size                      */
        unsigned long   total_buf_size;    /* total buffer size required       */
        unsigned short  num_entries;       /* number of entries                */
        unsigned short  total_num_entries; /* total number of entries          */
        unsigned char   list_options;      /* listing options                  */
        unsigned char   reserv3;           /* reserved                         */
        unsigned char   lu_name[8];        /* LU name                          */
        unsigned char   lu_alias[8];       /* LU alias                         */
        unsigned char   pu_name[8];        /* PU name filter                   */
} QUERY_LOCAL_LU;

typedef struct local_lu_summary
{
        unsigned short  overlay_size;      /* size of this entry               */
        unsigned char   lu_name[8];        /* LU name                          */
        unsigned char   lu_alias[8];       /* LU alias                         */
        unsigned char   description;       /* resource description             */
} LOCAL_LU_SUMMARY;

typedef struct local_lu_detail
{
        unsigned short  overlay_size;      /* size of this entry               */
        unsigned char   lu_name[8];        /* LU name                          */
        LOCAL_LU_DEF_DATA def_data;        /* defined data                     */
        LOCAL_LU_DEF_DATA det_data;        /* determined data                  */
} LOCAL_LU_DETAIL;
```

QUERY_LOCAL_LU

```
typedef struct local_lu_def_data
{
        unsigned char   description[RD_LEN];
                                        /* resource description            */
        unsigned char   lu_alias[8];    /* local LU alias                  */
        unsigned char   nau_address;    /* NAU address                     */
        unsigned char   syncpt_support; /* Reserved                        */
        unsigned short  lu_session_limit;  /* LU session limit             */
        unsigned char   default_pool;   /* member of default_lu_pool       */
        unsigned char   reserv2;        /* reserved                        */
        unsigned char   pu_name[8];     /* PU name                         */
        unsigned char   lu_attributes;  /* LU attributes                   */
        unsigned char   sscp_id[6];     /* SSCP ID                         */
        unsigned char   disable;        /* disable or enable Local LU      */
        unsigned char   attach_routing_data[128];
                                        /* routing data for                */
                                        /* incoming attaches               */
        unsigned char   lu_model;       /* LU model name for SDDLU         */
        unsigned char   model_name[8];  /* LU model name for SDDLU         */
        unsigned char   reserv4[16];    /* reserved                        */
} LOCAL_LU_DEF_DATA;

typedef struct local_lu_det_data
{
        unsigned char   lu_sscp_sess_active;
                                      /* Is LU-SSCP session active         */
        unsigned char   appl_conn_active; /* Is LU-SSCP session active     */
        unsigned char   reserv1[2];   /* reserved                          */
        SESSION_STATS   lu_sscp_stats;  /* LU-SSCP session statistics      */
        unsigned char   sscp_id[6];   /* SSCP ID                        */
} LOCAL_LU_DET_DATA;

typedef struct session_stats
{
        unsigned short  rcv_ru_size;    /* session receive RU size         */
        unsigned short  send_ru_size;   /* session send RU size            */
        unsigned short  max_send_btu_size; /* max send BTU size            */
        unsigned short  max_rcv_btu_size;  /* max rcv BTU size             */
        unsigned short  max_send_pac_win;  /* max send pacing win size     */
        unsigned short  cur_send_pac_win;  /* current send pacing win size */
        unsigned short  max_rcv_pac_win;   /* max receive pacing win size  */
        unsigned short  cur_rcv_pac_win;   /* current receive pacing       */
                                        /* window size                     */
        unsigned long   send_data_frames;  /* number of data frames sent   */
        unsigned long   send_fmd_data_frames;
                                        /* num of FMD data frames sent     */
        unsigned long   send_data_bytes;   /* number of data bytes sent    */
        unsigned long   rcv_data_frames;   /* num data frames received     */
        unsigned long   rcv_fmd_data_frames;
                                        /* num of FMD data frames recvd    */
        unsigned long   rcv_data_bytes;    /* number of data bytes received */
        unsigned char   sidh;           /* session ID high byte            */
        unsigned char   sidl;           /* session ID low byte             */
        unsigned char   odai;           /* ODAI bit set                    */
        unsigned char   ls_name[8];     /* Link station name               */
        unsigned char   pacing _type;   /* Type of pacing in use           */
} SESSION_STATS;
```

VCB a 构

格式 0

```
typedef struct local_lu_def_data
{
        unsigned char   description[RD_LEN];
                                        /* resource description            */
        unsigned char   lu_alias[8];    /* local LU alias                  */
```

```
        unsigned char   nau_address;       /* NAU address                      */
        unsigned char   syncpt_support;    /* Reserved                         */
        unsigned short  lu_session_limit;  /* LU session limit                 */
        unsigned char   default_pool;      /* member of default_lu_pool        */
        unsigned char   reserv2;           /* reserved                         */
        unsigned char   pu_name[8];        /* PU name                          */
        unsigned char   lu_attributes;     /* LU attributes                    */
        unsigned char   sscp_id[6];        /* SSCP ID                          */
        unsigned char   disable;           /* disable or enable Local LU       */
        unsigned char   attach_routing_data[128];
                                           /* routing data for                 */
                                           /* incoming attaches                */
} LOCAL_LU_DEF_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_LOCAL_LU

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? + ;会，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **lu_name** （或_，如果 **lu_name** h置为全c，则为 **lu_alias**）m > w引值， C w引值C Z指定要返回D 5际信息D起 < c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

AP_LIST_BY_ALIAS

返回D P m 4 **lu_alias** 排序。 v 1指定K AP_FIRST_IN_LIST 1， C

选项P效。如果指定K AP_LIST_FROM_NEXT 或AP_LIST_INCLUSIVE，那4Pm3序+取vZya供D **lu_name** 或**lu_alias** G否作为起<c。

lu_name

LU {F。C{FGv 8字ZD A型EBCDIC字符串。如果C字段h置为全c，则 **lu_alias** 字段+CZ确定w引。如果 **list_options** h置为AP_FIRST_IN_LIST，则忽TC字段。

lu_alias

>X定义D LU p{。|G在>XI显>字符集中D一v 8字Z字符串。yP 8 v字Z都GP效D，"RX须h置。如果 **lu_name** 和 **lu_alias** 字段都h置为全c，则9CkX制c（缺！ LU）相关*D LU。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽TC字段。

pu_name

PU {F过K器。|&Ch置为全c或为一v 8字Z字母}字D A型EBCDIC字符串（以一v字母*头），"以EBCDIC UqRnd。如果h置KC字段，那4只返回kC PU相关*D>X LU。 如果C字段h置为全c，则忽TC字段。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc

AP_OK

buf_size

在缓ex中返回D信息$度。

total_buf_size

返回值mw返回yP；ksDPm信息y需要D缓exD大小。C值I以大Z **buf_size**。

num_entries

5际返回D项D}?。

total_num_entries

已-返回D项D总}。C值I以大Z **num_entries**。

local_lu_summary.overlay_size

C项中D字Z}，即=下一v返回项（如果PD话）D偏移?。

local_lu_summary.lu_name

LU {F。C{FGv 8字ZD A型EBCDIC字符串。

local_lu_summary.lu_alias

>X定义D LU p{。|G在>XI显>字符集中D一v 8字Z字符串。yP D 8 v字ZGP效D。

local_lu_summary.description

资源5w（如DEFINE_LOCAL_LU中5wD）。|G一v以>XI显>字符集m>D 16字Z字符串。yP 16 v字Z都P效。

local_lu_detail.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移？。

local_lu_detail.lu_name

LU { F。C { F G v 8 字Z D A 型 EBCDIC 字符串。

local_lu_detail.def_data.description

资源5 w（如 DEFINE_LOCAL_LU 中5 w D）。|G一v以>X I显>字符集m>D 16 字Z字符串。y P 16 v字Z都P效。

local_lu_detail.def_data.lu_alias

>X定义D LU p{。|G在>X I显>字符集中D一v 8 字Z字符串。y P D 8 v字Z G P效D。

local_lu_detail.def_data.nau_address

LU D网g I寻址%元X址，范围在 0--255 内。非c值5指 LU G一从t LU。c值5指 LU G一独" LU。

local_lu_detail.def_data.syncpt_support

t。

local_lu_detail.def_data.lu_session_limit

>X LU D最大会话}？。c值m>无限制。

local_lu_detail.def_data.default_pool

如果 LU G从t LU 6.2 缺！X D一v I员，则h置 AP_YES。对Z独" LU，<终h置为 AP_NO。

local_lu_detail.def_data.pu_name

此 LU +9 C D PU D{ F。b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。此字段v I从t LU 9 C，R对独" LU +h置为全二x制c。

local_lu_detail.def_data.lu_attributes

已配置 LU t性。此字段I为值 AP_NONE，或下P值 OR（或Y作）组合:

AP_DISABLE_PWSUB

{ C>X LU D Z n f代支V。

local_lu_detail.def_data.sscp_id

此字段指定允许激活此 LU D SSCP D ID。|G一v 6 字Z二x制字段。此字段v I从t LU 9 C，R对独" LU、或如果 LU I任何 SSCP 激活&h置为全二x制c。

local_lu_detail.def_data.disable

C字段m w LOCAL LU &{ C还G启C。I通过重新发v带J 1 N }（AP_YES 或 AP_NO）D DEFINE_LOCAL_LU，动，X启C或{ C LU。1一v{ C D LU ；启C 1，L序+发v一v NOTIFY（*机）。1一v启C D LU ；{ C 1，L序+发v一v NOTIFY（脱机）。如果 LU ；{ C 1处Z，S状，，则L序+在 NOTIFY（脱机）之后发v一v UNBIND。

local_lu_detail.def_data.attach_routing_data

C字段m>I Z B务L序，S=C>X LU O，以致Z在 DYNAMIC_LOAD_INDICATION O发v D}]未|D。例如，C字段I能C Z h置一u=B务L序D工作目<D 7 6。

def_data.lu_model

LU D模型类型和号k。此字段v I从t LU 9C, R对独" LU &h置为AP_UNKNOWN。对Z从t LU, | h置为下P值之一:

AP_3270_DISPLAY_MODEL_2
AP_3270_DISPLAY_MODEL_3
AP_3270_DISPLAY_MODEL_4
AP_3270_DISPLAY_MODEL_5
AP_RJE_WKSTN
AP_PRINTER
AP_SCS_PRINTER
AP_UNKNOWN

对Z从t LU, 如果 **model_name** 未h置为全二x制c, 则忽T此字段。如果指定K } AP_UNKNOWN 外D其{值, R主机系统支V SDDLU(自定义从t LU), 则Z c + z I一v自动a供D PSID NMVT &答, 以动, X在主机O定义> X LU。PSID 子向? | 含k此字段相对&D机器类型和型号。通过重新发v动词, 此字段I动, X x行 | D。在 LU 关U和M放0, | D + ;起作C。

def_data.model_name

LU D模型{ F。此字段v I从t LU 9C, R对独" LU &h置为二x制c。

如果此字段未h置为二x制c, R主机系统支V SDDLU, 则Z c + z I一v自动a供D PSID NMVT &答, 以动, X在主机O定义> X LU。PSID 子向? | 含此字段中a供D{ F。通过重新发v动词, 此字段I动, X x行 | D。在LU 关U和M放0, | D + ;起作C。

local_lu_detail.det_data.lu_sscp_session_active

5 w LU-SSCP 会话G否G活动D (AP_YES 或 AP_NO)。如果 **def_data.nau_address** 为c, 那4 # t C字段。

local_lu_detail.det_data.appl_conn_active

指定&C L序G否9 C C LU(AP_YES 或 AP_NO)。 如果 **def_data.nau_address** 为c, 那4 # t C字段。

local_lu_detail.det_data.lu_sscp_stats.rcv_ru_size

<终# t C字段。

local_lu_detail.det_data.lu_sscp_stats.send_ru_size

<终# t C字段。

local_lu_detail.det_data.lu_sscp_stats.max_send_btu_size

能发MD最大 BTU 大小。

local_lu_detail.det_data.lu_sscp_stats.max_rcv_btu_size

能S U D最大 BTU 大小。

local_lu_detail.det_data.lu_sscp_stats.max_send_pac_win

<终+ C字段h置为c。

local_lu_detail.det_data.lu_sscp_stats.cur_send_pac_win

<终+ C字段h置为c。

local_lu_detail.det_data.lu_sscp_stats.max_rcv_pac_win

<终+ C字段h置为c。

local_lu_detail.det_data.lu_sscp_stats.cur_rcv_pac_win
< 终 + C 字段 h 置为 c 。

local_lu_detail.det_data.lu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ? 。

local_lu_detail.det_data.lu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ? 。

local_lu_detail.det_data.lu_sscp_stats.send_data_bytes
发M D } # }] w D字Z } 。

local_lu_detail.det_data.lu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ? 。

local_lu_detail.det_data.lu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ? 。

local_lu_detail.det_data.lu_sscp_stats.rcv_data_bytes
S U D } # }] w D字Z } 。

local_lu_detail.det_data.lu_sscp_stats.sidh
会话 ID _ 字Z 。

local_lu_detail.det_data.lu_sscp_stats.sidl
会话 ID M字Z 。

local_lu_detail.det_data.lu_sscp_stats.odai
起 < 目D X址指 > 符。 1 (" 会话1 , 如果 > X Z c | 含主 4 7 > , 则 ACTLU D发M方置C字段为0; " R如果 ACTLU D发M方G | 含从 4 7 > D Z c , 则置C字段为 1。

local_lu_detail.det_data.lu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P效D。 C字段能C Z Q会话D统计信息通过信息w ? 同 4 7 关 * 起来。

"： LU-SSCP 统计信息（**local_lu_detail.det_data.lu_sscp_stats**） v 1 **nau_address** 非 c 1 P效。否则 # t C字段。

local_lu_detail.det_data.lu_sscp_stats.pacing_type
LU-SSCP 会话中 } 在 9 C D S U w = 类型。 + h 置为 AP_NONE。

local_lu_detail.det_data.sscp_id
b G v | 含从C LU 9 C PU D ACTPU 中S U D SSCP ID D 6 字Z D字段。

如果因N } 错误而 9 动词 ; 能执行, 则L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LU_ALIAS

AP_INVALID_LU_NAME
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_LOCAL_TOPOLOGY

y P APPN Z c 维护 > X 拓扑 }] b , C }] b # 存关 Z = y P Z | Z c D 传 d 组 D 信息。

QUERY_LOCAL_TOPOLOGY 允许返回关 Z b 些 TG D 信息。

b 些信息以 * 要 q = 或详细信息 q = D P m 返回。如需关 Z 某 v X 定 > X TG D 信息，或 _ 获取几 v 『段』中 D P m 信息，& C h 置 **dest**、 **dest_type**、**tg_num** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T b 些字段。 k N 阅 Z 12 页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。 C P m W 先 4 **dest** x 行排序，然后 4 **dest_type** x 行排序，最后 4 **tg_num** 排序。 **dest** { F W 先 4 { F $ 度排序，然后对相同 $ 度 D { F 4 词 d ` 纂排序。 **dest_type** 4 下 P 次序排序: AP_LEN_NODE, AP_NETWORK_NODE, AP_END_NODE, AP_VRN。 **tg_num** 4 } 字大小排序。

如果已选 AP_LIST_INCLUSIVE，则返回 D P m 从 C { F D Z 一 v P 效记 < * < 。

如果选中 K AP_LIST_FROM_NEXT， P m + 从 { 为下 P 指定 { F 之一 D Z 一 v P 效记 < * < 。

VCB a 构

```
typedef struct query_local_topology
{
        unsigned short  opcode;            /* verb operation code        */
        unsigned char   reserv2;           /* reserved                   */
        unsigned char   format;            /* format                     */
        unsigned short  primary_rc;        /* primary return code        */
        unsigned long   secondary_rc;      /* secondary return code      */
        unsigned char   *buf_ptr;          /* pointer to buffer          */
        unsigned long   buf_size;          /* buffer size                */
        unsigned long   total_buf_size;    /* total buffer size required */
        unsigned short  num_entries;       /* number of entries          */
        unsigned short  total_num_entries; /* total number of entries    */
        unsigned char   list_options;      /* listing options            */
        unsigned char   reserv3;           /* reserved                   */
        unsigned char   dest[17];          /* TG destination node        */
        unsigned char   dest_type;         /* TG destination node type   */
        unsigned char   tg_num;            /* TG number                  */
} QUERY_LOCAL_TOPOLOGY;

typedef struct local_topology_summary
{
        unsigned short  overlay_size;      /* size of this entry         */
        unsigned char   dest[17];          /* TG destination node        */
        unsigned char   dest_type;         /* TG destination node type   */
        unsigned char   tg_num;            /* TG number                  */
} LOCAL_TOPOLOGY_SUMMARY;

typedef struct local_topology_detail
{
        unsigned short  overlay_size;      /* size of this entry         */
        unsigned char   dest[17];          /* TG destination node        */
        unsigned char   dest_type;         /* TG destination node type   */
        unsigned char   tg_num;            /* TG number                  */
        unsigned char   reserv1;           /* reserved                   */
        LINK_ADDRESS    dlc_data;          /* DLC signalling data        */
        unsigned long   rsn;               /* resource sequence number   */
```

```
        unsigned char   status;            /* TG status                  */
        TG_DEFINED_CHARS tg_chars;         /* TG characteristics         */
        unsigned char   cp_cp_session_active;
                                           /* CP-CP session is active    */
        unsigned char   branch_tg;         /* branch link type           */
        unsigned char   reserva[13];       /* reserved                   */
} LOCAL_TOPOLOGY_DETAIL;

typedef struct link_address
{
        unsigned short  length;            /* length                     */
        unsigned short  reserve1;          /* reserved                   */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                           /* address                    */
} LINK_ADDRESS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_LOCAL_TOPOLOGY

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ,过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要,过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回*要信息。

AP_DETAIL

返回详细信息。

y指定D **dest**、 **dest_type** 和 **tg_num** D组合（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

dest TG D全限定目D Z c {。{ F $度为17 v字Z， "以EBCDIC U q R n d。| I = v A型EBCDIC字符串组I，中间以一v EBCDIC c , S。(? v { F D $度最多I为8 v字Z， " ;含6入U q。）如果 **list_options** h置为AP_FIRST_IN_LIST，则忽T C字段。

dest_type
C TG D目D Z c D Z c类型。 | I h置为下P值之一:

AP_NETWORK_NODE
AP_VRN
AP_END_NODE
AP_LEARN_NODE

如果；知@ **dest_type**，则X须指定 AP_LEARN_NODE。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

tg_num
k TG 相关D号k。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回 y P ； k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ？。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**。

local_topology_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移？。

local_topology_summary.dest
TG D全限定目D Z c {。{ F $度为 17 v字Z， " 以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， " ；含6入U q。）

local_topology_summary.dest_type
C TG D目D Z c类型。 | h置为下P值之一:

AP_NETWORK_NODE
AP_VRN
AP_END_NODE

注意如果 **dest_type** h置为 AP_END_NODE， b + TG 目D X指定为 LEN Z c或一v末端Z c。

local_topology_summary.tg_num
k TG 相关D号k。

QUERY_LOCAL_TOPOLOGY

local_topology_detail.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移？。

local_topology_detail.dest
TG D全限定目D Z c {。{ F $度为 17 v字Z， " 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， " ；含6入U q。）

local_topology_detail.dest_type
C TG D目D Z c类型。| h置为下P值之一:

AP_NETWORK_NODE
AP_VRN
AP_END_NODE

注意如果 **dest_type** h置为 AP_END_NODE, b + TG 目D X指定为 LEN Z c或一v末端Z c。

local_topology_detail.tg_num
k TG 相关D号k。

local_topology_detail.dlc_data.length
= VRN D，S D DLC X址$度（如果 **dest_type** ；G AP_VRN, 则h置为c）。

local_topology_detail.dlc_data.address
= VRN D，S D DLC X址。

local_topology_detail.rsn
资源序P号。b I 5 P C资源D网g Z c分配。

local_topology_detail.status
指定 TG D状，。C N } I以G下P一v或多v和 OR（或Y作）一起D值:

AP_TG_OPERATIVE
AP_TG_CP_CP_SESSIONS
AP_TG_QUIESCING
AP_TG_HPR
AP_TG_RTP
AP_NONE

local_topology_detail.tg_chars
TG X性（N阅Z 35页D『DEFINE_CN』）。

local_topology_detail.cp_cp_session_active
指定>X Z c D：y $方 CP-CP 会话G否激活。

local_topology_detail.branch_link_type
v BrNN。C TG 分支4 7类型。| h置为下P之一:

AP_UPLINK
C 4 7 G一种O行4 7。

AP_DOWNLINK
C 4 7 G一种= EN D下行4 7。

AP_DOWNLINK_TO_BRNN
TG G一种显> | D EN S Z D = BrNN D下行４７。

AP_OTHERLINK
Ｃ４７G一种其｜４７。

其｜Ｚ c类型: C字段无意义R <终h置为 AP_BRNN_NOT_SUPPORTED。

local_topology_detail.branch_tg
v NN。５w TG G否G分支 TG。

AP_NO
TG ；G分支 TG。

AP_YES
TG G分支 TG。

其｜Ｚ c类型: C字段；P意义R一直h置为 AP_NO。

如果因N｝错误而9动词；能执行，则L序返回下P N｝:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_TG

AP_INVALID_LIST_OPTION

如果因Ｚ c P未启动而9动词；能执行，则L序返回下P N｝:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N｝:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_LS

QUERY_LS 返回关Z在Z c定义D 4 7 > D信息P m。C信息I 『确定 }] 』（执行期间动，Q集D }] ）和『定义 }] 』（DEFINE_LS O D & C L序y a供D }] ）构I。

b些信息以 * 要q = 或详细信息 q = D P m返回。如需P关某v X定 LS D信息，或要获C几v 『段』中D P m信息， & h置 **ls_name** 字段。

否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽T C字段。 k N阅Z 12页D 『 i 询Z c 』，以获取关Z如何9 C P m q = D 3 0知6。

C P m y] **ls_name** 排序。W先4 { F $度排序，然后对相同$度D { F x行 ASCII 词d ` 辑排序（ y] IBM D 6611 APPN MIB 定序）。如果选中K AP_LIST_FROM_NEXT，则返回D P m 4定义D次序（无 [指定D项G否存在）从下一项 * <。

要返回D 4 7 > D P m I 以 | G y t D端Z { F x行8选。在b种 i v下， & h置 **port_name** 字段（否则， C字段h置为全c）。

VCB a构

格式 1

```
typedef struct query_ls
{
        unsigned short  opcode;             /* verb operation code            */
        unsigned char   attributes;         /* Verb attributes                */
        unsigned char   format;             /* format                         */
        unsigned short  primary_rc;         /* Primary return code            */
        unsigned long   secondary_rc;       /* Secondary return code          */
        unsigned char   *buf_ptr;           /* pointer to buffer              */
        unsigned long   buf_size;           /* buffer size                    */
        unsigned long   total_buf_size;     /* total buffer size required     */
        unsigned short  num_entries;        /* number of entries              */
        unsigned short  total_num_entries;  /* total number of entries        */
        unsigned char   list_options;       /* listing options                */
        unsigned char   reserv3;            /* reserved                       */
        unsigned char   ls_name[8];         /* name of link station           */
        unsigned char   port_name[8];       /* name of link station           */
} QUERY_LS;

typedef struct ls_summary
{
        unsigned short  overlay_size;       /* size of this entry             */
        unsigned char   ls_name[8];         /* link station name              */
        unsigned char   description[RD_LEN];
                                            /* resource description           */
        unsigned char   dlc_type;           /* DLC type                       */
        unsigned char   state;              /* link station state             */
        unsigned short  act_sess_count;     /* currently active sess count    */
        unsigned char   det_adj_cp_name[17];
                                            /* determined adj CP name         */
        unsigned char   det_adj_cp_type;    /* determined adj node type       */
        unsigned char   port_name[8];       /* port name                      */
        unsigned char   adj_cp_name[17];    /* adjacent CP name               */
        unsigned char   adj_cp_type;        /* adjacent node type             */
} LS_SUMMARY;

typedef struct ls_detail
{
        unsigned short  overlay_size;       /* size of this entry             */
```

```
        unsigned char   ls_name[8];         /* link stations name            */
        LS_DET_DATA     det_data;           /* determined data               */
        LS_DEF_DATA     def_data;           /* defined data                  */
} LS_DETAIL;

typedef struct ls_det_data
{
        unsigned short  act_sess_count;     /* curr active sessions count    */
        unsigned char   dlc_type;           /* DLC type                      */
        unsigned char   state;              /* link station state            */
        unsigned char   sub_state;          /* link station sub state        */
        unsigned char   det_adj_cp_name[17];
                                            /* adjacent CP name              */
        unsigned char   det_adj_cp_type;    /* adjacent node type            */
        unsigned char   dlc_name[8];        /* name of DLC                   */
        unsigned char   dynamic;            /* is LS is dynamic ?            */
        unsigned char   migration;          /* supports migration partners   */
        unsigned char   tg_num;             /* TG number                     */
        LS_STATS        ls_stats;           /* link station statistics       */
        unsigned long   start_time;         /* time LS started               */
        unsigned long   stop_time;          /* time LS stopped               */
        unsigned long   up_time;            /* total time LS active          */
        unsigned long   current_state_time; /* time in current state         */
        unsigned char   deact_cause;        /* deactivation cause            */
        unsigned char   hpr_support;        /* TG HPR support                */
        unsigned char   anr_label[2];       /* local ANR label               */
        unsigned char   hpr_link_lvl_error; /* HPR link-level error          */
        unsigned char   auto_act;           /* auto activate                 */
        unsigned char   ls_role;            /* link station role             */
        unsigned char   reserva;            /* reserved                      */
        unsigned char   node_id[4];         /* determined node id            */
        unsigned short  active_isr_count;   /* currently active ISR sessions */
        unsigned short  active_lu_sess_count;
                                            /* active LU-LU session count    */
        unsigned short  active_sscp_sess_count;
                                            /* active SSCP session count     */
        ANR_LABEL       reverse_anr_labe;   /* reverse ANR label             */
        unsigned short  max_send_btu_size;  /* negotiated max BTU length     */
        unsigned char   brnn_link_type;     /* branch link type              */
        unsigned char   adj_cp_is_brnn;     /* adjacent CP is a BrNN         */
        unsigned char   reservb[6];        /* reserved                       */
} LS_DET_DATA;

typedef struct anr_label
{
        unsigned short  length;           /* ANR label length                */
        unsigned short  reserv;           /* reserved                        */
        unsigned char   label[MAX_ANR_LABEL_SIZE];
                                          /* ANR label                       */
} ANR_LABEL;

typedef struct ls_def_data
{
        unsigned char   description[RD_LEN];
                                            /* resource description          */
        unsigned char   port_name[8];       /* name of associated port       */
        unsigned char   adj_cp_name[17];    /* adjacent CP name              */
        unsigned char   adj_cp_type;        /* adjacent node type            */
        LINK_ADDRESS    dest_address;       /* destination address           */
        unsigned char   auto_act_supp;      /* auto-activate supported       */
        unsigned char   tg_number;          /* Pre-assigned TG number        */
        unsigned char   limited_resource;   /* limited resource              */
        unsigned char   solicit_sscp_sessions;
                                            /* solicit SSCP sessions         */
        unsigned char   pu_name[8];         /* Local PU name (reserved if    */
                                            /* solicit_sscp_sessions is set  */
                                            /* to AP_NO)                     */
        unsigned char   disable_remote_act; /* disable remote activation flag */
```

```
        unsigned char   dspu_services;      /* Services provided for          */
                                            /* downstream PU                  */
        unsigned char   dspu_name[8];       /* Downstream PU name (reserved   */
                                            /* if dspu_services is set to     */
                                            /* AP_NONE or AP_DLUR)            */
        unsigned char   dlus_name[17];      /* DLUS name if dspu_services     */
                                            /* is set to AP_DLUR              */
        unsigned char   bkup_dlus_name[17]; /* Backup DLUS name if            */
                                            /* dspu_services is set           */
                                            /* to AP_DLUR                     */
        unsigned char   hpr_supported;      /* does the link support HPR?     */
        unsigned char   hpr_link_lvl_error; /* does the link support HPR      */
                                            /* link-level error recovery?     */
        unsigned short  link_deact_timer;   /* HPR link deactivation timer    */
        unsigned char   reserv1;            /* reserved                       */
        unsigned char   default_nn_server;  /* Use as default LS to NN server */
        unsigned char   ls_attributes[4];   /* LS attributes                  */
        unsigned char   adj_node_id[4];     /* adjacent node ID               */
        unsigned char   local_node_id[4];   /* local node ID                  */
        unsigned char   cp_cp_sess_support; /* CP-CP session support          */
        unsigned char   use_default_tg_chars;
                                            /* Use default tg_chars           */
        TG_DEFINED_CHARS tg_chars;          /* TG characteristics             */
        unsigned short  target_pacing_count;
                                            /* target pacing count            */
        unsigned short  max_send_btu_size;  /* max send BTU size              */
        unsigned char   ls_role;            /* link station role to use       */
                                            /* on this link                   */
        unsigned char   max_ifrm_rcvd;      /* max number of I-frames rcvd    */
        unsigned short  dlus_retry_timeout; /* DLUS retry timeout             */
        unsigned short  dlus_retry_limit;   /* DLUS retry limit               */
        unsigned char   conventional_lu_compression;
                                            /* Data compression requested for */
                                            /* conventional LU sessions       */
        unsigned char   conventional_lu_cryptography;
                                            /* Cryptography required for      */
                                            /* conventional LU sessions       */
        unsigned char   reserv3;            /* reserved                       */
        unsigned char   retry_flags;        /* conditions for automatic       */
                                            /* retries                        */
        unsigned short  max_activation_attempts;
                                            /* how many automatic retries:    */
        unsigned short  activation_delay_timer;
                                            /* delay between automatic        */
                                            /* retries                        */
        unsigned char   branch_link_type;   /* branch link type               */
        unsigned char   adj_brnn_cp_support;/* adjacent BrNN CP support       */
        unsigned char   reserv4[20];        /* reserved                       */
        unsigned short  link_spec_data_len; /* length of link specific data   */
} LS_DEF_DATA;

typedef struct link_address
{
        unsigned short  length;             /* length                         */
        unsigned short  reserve1;           /* reserved                       */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                            /* address                        */
} LINK_ADDRESS;

typedef struct link_spec_data
{
       unsigned char  link_data[SIZEOF_LINK_SPEC_DATA];

}  LINK_SPEC_DATA;

typedef struct tg_defined_chars
{
        unsigned char   effect_cap;         /* Effective capacity             */
```

```
        unsigned char   reserve1[5];        /* Reserved                        */
        unsigned char   connect_cost;       /* Connection Cost                 */
        unsigned char   byte_cost;          /* Byte cost                       */
        unsigned char   reserve2;           /* Reserved                        */
        unsigned char   security;           /* Security                        */
        unsigned char   prop_delay;         /* Propagation delay               */
        unsigned char   modem_class;        /* Modem class                     */
        unsigned char   user_def_parm_1;    /* User-defined parameter 1        */
        unsigned char   user_def_parm_2;    /* User-defined parameter 2        */
        unsigned char   user_def_parm_3;    /* User-defined parameter 3        */
} TG_DEFINED_CHARS;

typedef struct ls_stats
{
        unsigned long   in_xid_bytes;       /* number of XID bytes received    */
        unsigned long   in_msg_bytes;       /* num message bytes received      */
        unsigned long   in_xid_frames;      /* num XID frames received         */
        unsigned long   in_msg_frames;      /* num message frames received     */
        unsigned long   out_xid_bytes;      /* num XID bytes sent              */
        unsigned long   out_msg_bytes;      /* num message bytes sent          */
        unsigned long   out_xid_frames;     /* num XID frames sent             */
        unsigned long   out_msg_frames;     /* num message frames sent         */
        unsigned long   in_invalid_sna_frames;
                                            /* num invalid frames received     */
        unsigned long   in_session_control_frames;
                                            /* num control frames received     */
        unsigned long   out_session_control_frames;
                                            /* num control frames sent         */
        unsigned long   echo_rsps;          /* response from adj LS count      */
        unsigned long   current_delay;      /* time taken for last test sig    */
        unsigned long   max_delay;          /* max delay by test signal        */
        unsigned long   min_delay;          /* min delay by test signal        */
        unsigned long   max_delay_time;     /* time since longest delay        */
        unsigned long   good_xids;          /* successful XID on LS count      */
        unsigned long   bad_xids;           /* unsuccessful XID on LS count    */
} LS_STATS;
```

VCB a 构

格式 0（后备 6 别）

```
typedef struct ls_det_data
{
        unsigned short  act_sess_count;     /* curr active sessions count      */
        unsigned char   dlc_type;           /* DLC type                        */
        unsigned char   state;              /* link station state              */
        unsigned char   sub_state;          /* link station sub state          */
        unsigned char   det_adj_cp_name[17];
                                            /* adjacent CP name                */
        unsigned char   det_adj_cp_type;    /* adjacent node type              */
        unsigned char   dlc_name[8];        /* name of DLC                     */
        unsigned char   dynamic;            /* is LS is dynamic ?              */
        unsigned char   migration;          /* supports migration partners     */
        unsigned char   tg_num;             /* TG number                       */
        LS_STATS        ls_stats;           /* link station statistics         */
        unsigned long   start_time;         /* time LS started                 */
        unsigned long   stop_time;          /* time LS stopped                 */
        unsigned long   up_time;            /* total time LS active            */
        unsigned long   current_state_time; /* time in current state           */
        unsigned char   deact_cause;        /* deactivation cause              */
        unsigned char   hpr_support;        /* TG HPR support                  */
        unsigned char   anr_label[2];       /* local ANR label                 */
        unsigned char   hpr_link_lvl_error; /* HPR link-level error            */
        unsigned char   auto_act;           /* auto activate                   */
        unsigned char   ls_role;            /* link station role               */
        unsigned char   reserva;            /* reserved                        */
```

```
        unsigned char   node_id[4];         /* determined node id             */
        unsigned short  active_isr_count;   /* currently active ISR sessions  */
        unsigned char   reservb[30];        /* reserved                       */
} LS_DET_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_LS

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+此字段h置为 1，以指定O f P v D VCB D q = 1 f >。如果C字段h置为 0，L序返回q = 0 LS_DET_DATA a构。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。c值m > 无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **ls_name**（见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

ls_name

4 7 > { F。|G在> X I显> 字符集中D一v 8 字Z字符串。y P 8 v字Z都G P效D，" R X须h置。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

port_name

端Z { F过K器。 b & h置为 0 或 _ 一 v 8 字Z D字母 } 字集D A 型EBCDIC 字符串（以字母 * < ）， I EBCDIC U q R n d。如果 h 置K C字段，那 4 只返回 t Z C端Z D 4 7 > 。 如果C字段 h 置为全 c ，则忽T C字段。

返回N数

如果动词 I 功执行，则L 序返回下 P N } :

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ； k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大Z **buf_size**。

num_entries

5 际返回D项D } ？。

total_num_entries

已 - 返回D项D总 } 。 C值 I 以大Z **num_entries**。

ls_summary.overlay_size

C项中D字Z } ， 即 = 下一 v 返回项（如果 P D话）D偏移？。

ls_summary.ls_name

4 7 > D { 字。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v 字Z G P 效D。

ls_summary.description

资源 5 w（如在 DEFINE_LS 中 5 w D ）。 | G一 v 以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v 字Z 都P 效。

ls_summary.dlc_type

DLC D类型。 v 人通信或通信服务器支V下 P 类型:

AP_ANYNET
AP_LLC2
AP_OEM_DLC
AP_SDLC
AP_TWINAX
AP_X25

= 加D DLC 类型 I 以通过在 DEFINE_DLC 动词O指定新D类型来定义。 k N阅Z 49页D『DEFINE_DLC』， 以获取 | 详细D信息。

ls_summary.state

C 4 7 > D状，。 C字段 h 置为下 P 值之一:

AP_NOT_ACTIVE
AP_PENDING_ACTIVE
AP_ACTIVE
AP_PENDING_INACTIVE

ls_summary.act_sess_count
9 C 4 7 D活动会话总？（端 c 和 = i ）。

ls_summary.det_adj_cp_name
在 4 7 激活期间确定D 17 字 Z D全限定 Z | CP { F。 | I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串组 I， R I EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v字 Z， " ；含 6 人U q。）如果 LS G非活动D， b + G U D。

如果 **ls_summary.adj_cp_type** ；G AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE 或 AP_BACK_LEVEL_LEN_NODE，则 # t C字段。

ls_summary.det_adj_cp_type
在 4 7 激活期间确定D Z | Z c D类型。 | G下 P 值之一:

AP_END_NODE
AP_NETWORK_NODE
AP_LEARN_NODE
AP_VRN

如果 LS G非活动D， b + G AP_LEARN_NODE。

如果 **ls_summary.adj_cp_type** ；G AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE 或 AP_BACK_LEVEL_LEN_NODE，则 # t C字段。

ls_summary.port_name
k 此 4 7 > 关 * D端 Z D { F。 | G在 > X I 显 > 字符集中D一 v 8 字 Z 字符串。 y P D 8 v字 Z G P 效D。

ls_summary.adj_cp_name
b G v C EBCDIC c 串 * D = v A 型 EBCDIC 字符串组 I D， I EBCDIC U q R n d D 17 字 Z 全限定 Z | X制 c { F。（？v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人U q。）对 Z 隐 = 4 7， b G U D。

ls_summary.adj_cp_type
Z | Z c D类型。 | G下 P 值之一:

AP_END_NODE
AP_NETWORK_NODE
AP_APPN_NODE
AP_BACK_LEVEL_LEN__NODE
AP_HOST_XID3
AP_HOST_XID0
AP_DSPU_XID
AP_DSPU_NOXID

ls_detail.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移？。

ls_detail.ls_name

４７>D{字。｜G在>X I显>字符集中D一v ８字Z字符串。y P D ８ v字Z G P效D。

ls_detail.det_data.act_sess_count

９C４７D活动会话总？ (端c和=i)。

ls_detail.det_data.dlc_type

DLC D类型。v人通信或通信服务器支V下P类型:

AP_ANYNET
AP_LLC2
AP_OEM_DLC
AP_SDLC
AP_TWINAX
AP_X25

=加D DLC 类型I以通过在 DEFINE_DLC 动词O指定新D类型来定义。kN阅Z 49页D『DEFINE_DLC』，以获取｜详细D信息。

ls_detail.det_data.state

C４７>D状，。C字段h置为下P值之一:

AP_NOT_ACTIVE
AP_PENDING_ACTIVE
AP_ACTIVE
AP_PENDING_INACTIVE

ls_detail.det_data.sub_state

C字段a供K关Z C４７>D状，D｜详细D信息。C字段h置为下P值之一:

AP_SENT_CONNECT_OUT
AP_PENDING_XID_EXCHANGE
AP_SENT_ACTIVATE_AS
AP_SENT_SET_MODE
AP_ACTIVE
AP_SENT_DEACTIVATE_AS_ORDERLY
AP_SENT_DISCONNECT
AP_WAITING_STATS
AP_RESET

ls_detail.det_data.det_adj_cp_name

在４７激活期间确定D 17字Z D全限定Z｜X制c{F。｜I=v C EBCDIC c串*D A型EBCDIC字符串组I，R I EBCDIC U q R n d。(？v{F D$度最多I为８v字Z，"；含6入U q。)

如果 **ls_summary.adj_cp_type** ；G AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE 或 AP_BACK_LEVEL_LEN_NODE，则#t C字段。

ls_detail.det_data.det_adj_cp_type
在４７激活期间确定ＤＺ｜Ｚｃ Ｄ类型。｜Ｇ下Ｐ值之一:

AP_END_NODE
AP_NETWORK_NODE
AP_LEARN_NODE
AP_VRN

如果 **ls_summary.adj_cp_type** ；Ｇ AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE 或 AP_BACK_LEVEL_LEN_NODE，则#ｔＣ字段。

ls_detail.det_data.dlc_name
DLC Ｄ｛Ｆ。｜Ｇ在>ＸＩ显>字符集中Ｄ一ｖ ８字Ｚ字符串。ｙＰＤ８ｖ字ＺＧＰ效Ｄ。

ls_detail.det_data.dynamic
指定（Ｃ DEFINE_LS ｜ｎ）显=、隐=还Ｇ动，（响&来自Ｚ｜ＺｃＤ，Ｓｋｓ，或_ｇ越，Ｓ网ｇ动，，Ｓ=ｍ一ｖＺｃ）定义４７。Ｉｈ置为 AP_YES 或 AP_NO。

ls_detail.det_data.migration
指定Ｚ｜ＺｃＧ(移级Ｚｃ（如Ｍ入Ｚ，网（LEN）Ｚｃ）、全 APPN 网ｇＺｃ还Ｇ末端Ｚｃ（AP_YES、AP_NO 或 AP_UNKNOWN)。

ls_detail.det_data.tg_num
ｋ TG 相关Ｄ号ｋ。

ls_detail.det_data.ls_stats.in_xid_bytes
Ｃ４７>ＳＵＤ XID（；换ｊ ６）Ｄ字Ｚ总？。

ls_detail.det_data.ls_stats.in_msg_bytes
Ｃ４７>ＳＵＤ｝］字Ｚ总？。

ls_detail.det_data.ls_stats.in_xid_frames
Ｃ４７>ＳＵＤ XID (；换ｊ ６)Ｄ总？。

ls_detail.det_data.ls_stats.in_msg_frames
Ｃ４７>ＳＵＤ｝］帧Ｄ总？。

ls_detail.det_data.ls_stats.out_xid_bytes
Ｃ４７>发ＭＤ XID (；换ｊ ６符)Ｄ总字Ｚ｝。

ls_detail.det_data.ls_stats.out_msg_bytes
Ｃ４７>发ＭＤ｝］Ｄ总字Ｚ｝。

ls_detail.det_data.ls_stats.out_xid_frames
Ｃ４７>发ＭＤ XID (；换ｊ ６符)帧Ｄ总？。

ls_detail.det_data.ls_stats.out_msg_frames
Ｃ４７>发ＭＤ｝］帧Ｄ总？。

ls_detail.det_data.ls_stats.in_invalid_sna_frames
Ｃ４７>ＳＵＤ；｝确Ｄ SNA 帧Ｄ总？。

ls_detail.det_data.ls_stats.in_session_control_frames
Ｃ４７>ＳＵＤ会话Ｘ制帧Ｄ总？。

ls_detail.det_data.ls_stats.out_session_control_frames
C 4 7 > 发M D 会话X 制帧D 总？。

ls_detail.det_data.ls_stats.echo_rsps
从相Z Z c S U D & 答。周期性D 发M & 答 k s 以规定 = 相Z Z c D 传% 延 1 。

ls_detail.det_data.ls_stats.current_delay
从 b v 4 7 > = Z S 4 7 > 发M 和返回 D 最 Y D b T 信号 y 花 D 1 间（毫 k)。

ls_detail.det_data.ls_stats.max_delay
从C 4 7 > = Z S 4 7 > 发M 和返回 b T 信号 y 花D 最 $ 1 间(毫 k)。

ls_detail.det_data.ls_stats.min_delay
从C 4 7 > = Z S 4 7 > 发M 和返回 b T 信号 y 花D 最短 1 间(毫 k)。

ls_detail.det_data.ls_stats.max_delay_time
1 延Y 发 z 1 ，从系统启动 * < D 1 间(以Y 分之一 k 为% 位)。

ls_detail.det_data.ls_stats.good_xids
从 4 7 > 启动起，发 z 在 4 7 > O D I 功D XID ；换D 总？。

ls_detail.det_data.ls_stats.bad_xids
从 4 7 > 启动起，发 z 在 4 7 > O D ； I 功D XID ；换D 总？。

ls_detail.det_data.start_time
1 4 7 > O 一次激活 1 （即方 = h 置 | n 完 I ），从系统启动 * < D 1 间 (以Y 分之一 k 为% 位)。

ls_detail.det_data.stop_time
1 4 7 > O 一次M 放 1 ，从系统启动 * < D 1 间 (以Y 分之一 k 为% 位)。

ls_detail.det_data.up_time
自从系统启动后， 4 7 > 处Z 活动状，D 总D 1 间（以Y 分之一 k 为% 位）。

ls_detail.det_data.current_state_time
4 7 > 处Z 1 0 状，D 总D 1 间（以Y 分之一 k 为% 位）。

ls_detail.det_data.deact_cause
4 7 > O 一次M 放D 原因。字段 h 置为下 P 值之一:

AP_NONE
4 7 > 从未；M 放。

AP_DEACT_OPER_ORDERLY
C Z Y 作员发 v P 序 STOP | n， 4 7 > ；M 放。

AP_DEACT_OPER_IMMEDIATE
C Z Y 作员发 v " 即 STOP | n， 4 7 > ；M 放。

AP_DEACT_AUTOMATIC
4 7 > ；自动例如，例如，因为；再 P 会话 9 C 4 7 > 。

AP_DEACT_FAILURE
I Z 故O 4 7 > ；M 放。

ls_detail.det_data.hpr_support
在C TG O \ 支V D _ 性能 7 I 选择（HPR ）D 级 p （AP_NONE、AP_BASE 或 AP_RTP ）， | < G = > X 和Z | Z c D 能&。

ls_detail.det_data.anr_label
分配 = > X 4 7 D HPR 自动网 g 7 I 选择（ANR） j 号。

ls_detail.det_data.hpr_link_lvl_error
指定 4 S 级 p 错误恢 4 G 否 C Z 4 7 O D HPR 通信？。

ls_detail.def_data.auto_act
指定 4 7 1 0 G 否允许远 L 激活器 4 要 s 激活。返回下 P 值（或下 P 值 OR（或 Y 作）组合）：

AP_AUTO_ACT
4 7 I I > X Z c 4 要 s 激活。

AP_REMOTE_ACT
4 7 I I 远 L Z c 4 要 s 激活。

ls_detail.det_data.ls_role
C 4 7 > D 4 7 > G + 。起 u h 置为对 4 7 > 定义 D 4 7 > G + 。如果 y 定义 D G + G I 协 L D， C 值在 XID ；换期间 D 为协 L D G +（主或次）， " R 1 4 7 ；M 放 1， 转回 I 协 L D。

AP_LS_NEG
4 7 > G + G I 协 L D。

AP_LS_PRI
4 7 > G + 为主。

AP_LS_SEC
4 7 > G + 为次。

ls_detail.det_data.node_id
在 XID ；换期间从 Z | Z c S U D Z c ID。 | G 一 v 4 字 Z . y x 制字符串。

ls_detail.det_data.active_isr_count
9 C 4 7 D 活动 D 中间会话 D } ？。

ls_detail.det_data.active_lu_sess_count
9 C 4 7 D 活动 D LU-LU 会话 } ？。

ls_detail.det_data.active_sscp_sess_count
9 C 4 7 D 活动 D LU-SSCP 和 PU-SSCP 会话 } ？。

ls_detail.det_data.reverse_anr_label.length
4 7 > D 反向自动网 g 7 I 选择（anr） j 号 D $ 度。如果 4 7 ；支 V HPR， 或 _ j 号为知， C 字段 h 置为 c 。

ls_detail.det_data.local_address
C 4 7 > D > X X 址。

ls_detail.det_data.max_send_btu_size
I 在此 4 7 O 发 M D 最大 BTU 大小， 通过 k Z | Z c 协 L 来确定。如果未 " T 4 7 激活， 返回 c 。

ls_detail.det_data.brnn_link_type
v BrNN。C 分支 4 7 类型， | G 下 P 之一：

AP_UPLINK
C 4 7 G 一种 O 行 4 7 。

AP_DOWNLINK
4 7 为下行 4 7。

AP_OTHERLINK
C 4 7 G 一种其 | 4 7。

AP_UNKNOWN_LINK_TYPE
C 4 7 G 一种其 | 4 7。

AP_BRNN_NOT_SUPPORTED
C 4 7 只支 V PU 2.0 通信？。

其 | Z c 类型: C 字段无意义 R < 终 h 置为 AP_BRNN_NOT_SUPPORTED。

ls_detail.det_data.adj_cp_is_brnn
y P Z c 类型: 指定相 Z Z c G 否为一 v BRNN。

AP_UNKNOWN
；确定指定相 Z Z c G 否为一 v BRNN。

AP_NO
相 Z Z c ；G 一 v BrNN。

AP_YES
相 Z Z c G BrNN。

ls_detail.def_data.description
资源 5 w（如在 DEFINE_LS 中 5 w D）。 | G 一 v 以 > X I 显 > 字符集 m > D 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

ls_detail.def_data.port_name
k 此 4 7 > 关 * D 端 Z D { F。 | G 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。 y P D 8 v 字 Z G P 效 D。如果 4 7 ，S = VRN, 则 C 字段指定 C Z ，S VRN D 5 际端 Z { F（在 DEFINE_CN 动词中指定）。

ls_detail.def_data.adj_cp_name
17 字 Z D 全限定 Z | X 制 c { F，此 { F I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S， " 以 EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 入 U q。）如果 **back_lvl_len_end_node** 未 h 置为 AP_NO, 或 _ k 4 7 > 关 * D D 端 Z 定义为；换 =，则定义 C { F。

ls_detail.def_data.adj_cp_type
相 Z Z c 类型。

AP_NETWORK_NODE
指定 Z c 为 APPN 网 g Z c。

AP_END_NODE
指定 Z c 为 APPN 端 Z c 或外 c LEN Z c。

AP_APPN_NODE
指定 Z c 为 APPN 网 g Z c、APPN 端 Z c 或外 c LEN Z c。Z c 类型 + 在 XID ；换期间获 C。

AP_BACK_LEVEL_LEN_NODE
指定 Z c 为内 c LEN Z c。

AP_HOST_XID3
指定Z c为一v主机, " R Z c运c符h)对来自Z c（ q = 3 XID）D V询 XID 做v响&。

AP_HOST_XID0
指定Z c为一v主机, " R Z c运c符h)对来自Z c（ q = 0 XID）D V询 XID 做v响&。

AP_DSPU_XID
指定Z c为下N PU, " R Z c运c符h) + XID ；换|括在4 7激活中。

AP_DSPU_NOXID
指定Z c为下N PU, " R Z c运c符h) + XID ；换|括在4 7激活中。

": 一v至 VRN D4 7 > <终G动，D，因此未定义。

ls_detail.def_data.dest_address.length
相Z Z c O目D4 7 > X址D $度。

ls_detail.def_data.dest_address.address
相Z Z c O4 7 > D目D X址。

ls_detail.def_data.auto_act_supp
指定4 7在C START_LS 动词启动以及C STOP_LS 停止之后, G否自动激活。（AP_YES 或 AP_NO)。

ls_detail.def_data.tg_number
预先分配D TG 号（范围在 1 = 20)。1 4 7激活1此}字C Z m > 4 7。 c m > TG }字；G预先指定D, "在4 7激活1 x行协L。

ls_detail.def_data.limited_resource
指定1；P会话9 C 4 7 1, G否M放4 7 >。 | h置为下P值之一:

AP_NO
4 7；G一v \限资源, "；能自动M放。

AP_YES or AP_NO_SESSIONS
4 7 G一v \限资源, " 1；P活动会话9 C | 1 +自动M放。

AP_INACTIVITY
4 7 G一v \限资源, 1；P活动会话9 C | 1, 或1在指定1间内（I **link_deact_timer** 字段指定）4 7 O；P }] w动1, 4 7 +自动M放。

ls_detail.def_data.solicit_sscp_sessions
AP_YES k s主机启动 SSCP k > X X制c及从t LU 之间D会话。 AP_NO k s在此4 7；P k SSCP D会话。

ls_detail.def_data.pu_name
如果 **solicit_sscp_sessions** h置为 AP_YES, + 9 C C 4 7 D > X PU D { F。b G一v 8 字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头）, "以 EBCDIC U q R n d。如果 **solicit_sscp_sessions** h置为 AP_NO, # t C字段。

ls_detail.def_data.disable_remote.act

指定G否支V此４７D远L激活（AP_YES 或 AP_NO）。

ls_detail.def_data.dspu_services

如果 **solicit_sscp_sessions** h置为 AP_NO，指定＞X Z c在４７中a供x下N PU D服务。｜h置为下P值之一:

AP_PU_CONCENTRATION

＞X Z c＋向下N PU a供 PU 集中。

AP_DLUR

＞X Z c＋向下N PU a供 DLUR 服务。

AP_NONE

＞X Z c；向下N PU a供任何服务。

如果 **solicit_sscp_sessions** h置为 AP_YES，＃ t C字段。

ls_detail.def_data.dspu_name

下N PU D｛F。b G一v 8字Z字母｝字D A 型 EBCDIC 字符串（以一v字母＊头），"以 EBCDIC U q R n d。v 1 **solicit_sscp_sessions** h置为 AP_NO，C字段P效。

ls_detail.def_data.dlus_name

１至下N Z c D４７激活１，DLUR 向其k s SSCP 服务D DLUS Z c D｛F。C值h置为全c，或I＝v A 型 EBCDIC 字符串组I、中间以一v EBCDIC c，S、"以 EBCDIC U q R n d D 17 字Z字符串。（？v｛F D $度最多I为８v字Z，"；含６入U q。）如果此字段h置为全c，则１４７激活１全V缺！ DLUS（如果９C DEFINE_DLUR_DEFAULTS 动词定义）＋；k s。如果 **dlus_name** h置为c，"R无全V缺！ DLUS，则１４７激活１ DLUR ＋；启动 SSCP ＊系。如果 **dspu_services** 未h置为 AP_DLUR，则＃ t C字段。

ls_detail.def_data.bkup_dlus_name

作为下N PU ８份９C D DLUS Z c D｛F。C值h置为全c，或I＝v A型 EBCDIC 字符串组I、中间以一v EBCDIC c，S、"以 EBCDIC U q R n d D 17 字Z字符串。（？v｛F D $度最多I为８v字Z，"；含６入U q。）如果此字段h置为全c，则全V８份缺！ DLUS（如果９C DEFINE_DLUR_DEFAULTS 动词定义）＋作为此 PU D８份９C。如果 **dspu_services** 未h置为 AP_DLUR，则＃ t C字段。

ls_detail.def_data.hpr_supported

指定在此４７O G否支V HPR（AP_YES 或 AP_NO）。

ls_detail.def_data.hpr_link_lvl_error

指定在此４７O G否支V HPR ４S级p错误恢４～（AP_YES 或 AP_NO）。注意，如果 **hpr_supported** h置为 AP_NO，＃ t此N｝。

ls_detail.def_data.link_deact_timer

＼限资源M放计１器（k）。

如果 **limited_resource** h置为 AP_YES 或 AP_NO_SESSIONS，则如果在C计１器期间；P｝］在４７O穿过，４７＋自动M放，R无会话９C４７。

如果 **limited_resource** h置为 AP_INACTIVITY，则如果在C计１器期间；P｝］在４７O穿过，４７＋自动M放。

ls_detail.def_data.default_nn_server

指定 4 7 G否 I I 一 v 端 Z c 自动激活，以支 V 至网 g Z c 服务器 D CP-CP 会话。（AP_YES 或 AP_NO）。4 7 X 须定义为支 V CP-CP 会话，以 9 此字段 z 效。

ls_detail.def_data.ls_attributes

指定 P 关相 Z Z c D x 一 = 信息。

ls_detail.def_data.ls_attributes[0]

主机类型。

AP_SNA

j 准 SNA 主机。

AP_FNA

FNA（VTAM-F）主机。

AP_HNA

HNA 主机。

def_data.ls_attributes[1]

b G 一 v 位字段。| I 能 9 C 值 AP_NO 或任何下 P 4 位 OR（或 Y 作）组合:

AP_SUPPRESS_CP_NAME

至内 c LEN Z c 4 7 D网 g { F CV 抑制选项。如果 h 置此位，则在 k 相 Z Z c D XID ；换中；| 含网 g { F CV。（ } 非 **adj_cp_type** h 置为 AP_BACK_LEVEL_LEN_NODE 或 AP_HOST_XID3，否则此位 + ；忽 T。）

AP_REACTIVATE_ON_FAILURE

如果 4 7 G 活动 D + f 即 ' \， v 人通信或通信服务器 + " T 重新激活 4 7。如果重新激活 4 7 " T ' \，则 4 7 + # V 非活动。

AP_USE_PU_NAME_IN_XID_CVS

如果相 Z Z c ；定义为一 v 主机或在至 APPN Z c 4 7 O D **solicit_sscp_sessions** ；h 置为 AP_YES，R AP_SUPPRESS_CP_NAME 位未 h 置，则以 q = 3 XID 发 M D网 g { F CV 中 D 全限定 CP { F + I **def_data.pu_name** 中 a 供 D { F（以 CP D 网 g ID 全限定）代 f 。

ls_detail.def_data.adj_node_id

y 定义 D 相 Z Z c D Z c ID。

ls_detail.def_data.local_node_id

4 7 > O 以 XID 形 = 发 M D Z c ID。b G v 4 字 Z D . y x 制字符串。如果 C 字段 h 置为 0，**node_id** 在 XID ；换中 9 C。如果此字段非 c，则 | + 代 f 此 LS O XID ；换 D 值。

ls_detail.def_data.cp_cp_sess_support

指定 G 否支 V CP-CP 会话（AP_YES 或 AP_NO）。

ls_detail_def_data.use_default_tg_chars

指定废弃在 DEFINE_LS O a 供 D TG X 性而支 V DEFINE_PORT 动词 O a 供 D 缺！ TG X 性（AP_YES 或 AP_NO）。C 字段；J C Z 隐 = 4 7。

ls_detail.def_data.tg_chars

TG X 性（N 阅 Z 35 页 D 『DEFINE_CN』）。

ls_detail.def_data.target_pacing_count

1 和 32 767 之间D } 字值（ | 括 1 和 32 767），指 > 此 TG O BIND D期望w = 窗Z大小。 v 1 执行固定，S w = 1，此 } 字E P效。注意， v 人通信或通信服务器 1 0 ； 9 C此值。

ls_detail.def_data.max_send_btu_size

能发M D最大 BTU D大小。

ls_detail.def_data.ls_role

此 4 7 > & ；假定为D 4 7 > G + 。 | I 为 AP_LS_NEG、AP_LS_PRI 或 AP_LS_SEC 中D任一 v，以选择 I 协L、主或 (助。此字段也 I h 置为 AP_USE_PORT_DEFAULTS，以选择在 DEFINE_PORT 动词O配置D值。

ls_detail.def_data.max_ifrm_rcvd

XID 发M方在确认 0 I S U = D最大 I-帧 } 目。如果要 9 C来自 DEFINE_PORT D缺！值，则C字段 h 置为 0。

ls_detail.def_data.dlus_retry_timeout

在 Z 二次和后继 " T k 在 **ls_detail.def_data.dlus_name** 和 **ls_detail.def_data.bkup_dlus_name** 字段中指定D DLUS 取C * 系间D间 t （ k ）。u < " T和Z一次重T之间D间 t < 终为 1 k。如果指定 c，则 + 9 C 通过 DEFINE_DLUR_DEFAULTS 配置D缺！值。如果 **def_data.dspu_services** 未 h 置为 AP_DLUR，则忽T此字段。

ls_detail.def_data.dlus_retry_limit

在 Z 一次 ' \ 后继续 k 在 **ls_detail.def_data.dlus_name** 和 **ls_detail.def_data.bkup_dlus_name** 字段中指定D DLUS 取C * 系D最大重T次 } 。如果指定 c，则 + 9 C通过 DEFINE_DLUR_DEFAULTS 配置D缺！值。如果指定 X'FFFF'，则 L 序 + ；确定 X x 行重 T 。如果 **def_data.dspu_services** 未 h 置为 AP_DLUR，则忽T此字段。

ls_detail.def_data.link_spec_data_len

在 u < 化期间， }] D未 n z $ 度（以字Z为%位），C }] 未做 | D传M = 4 7 > 组件。 }] " 置 = LS_DETAIL a 构中。C }] + n z 4 字Z范围。

ls_detail.def_data.convention_lu_compression

指定对C 4 7 O D会话G否 k s }] 压 u。注意，此字段 v 对其O P 载 LU 0 = 3 通信？D 4 7 P 效。

AP_NO

> X Z c ；压 u 或 b 压在C 4 7 O w 动D # 规 LU }] 。

AP_YES

如果主机 k s 对 }] x 行压 u，则 }] 压 u & 对C 4 7 O D # 规 LU 会话启C。

ls_detail.def_data.convention_lu_cryptography

指定 # 规 LU 会话G否要 s 会话 c 加 \ 。此字段 v J C Z 其O P 载 # 规 LU 通信？D 4 7 。

AP_NONE

会话 c 加 \ ； I L 序执行。

AP_MANDATORY

如果 LU P一<入\钥，则?制性会话c加\ I L序执行。否则，|X须I 9C LU D&C L序执行（如果b G PU 集中，则I下N LU 执行)。

AP_OPTIONAL

此值允许I主机&C L序以?会话为基础来}动加\。如果主机k s对一v k此 PU 关* D会话x行加\，则L序D行为X w k AP_MANDATORY 相同。如果主机未k s加\，则其行为X w k AP_NONE 相同。

ls_detail.def_data.retry_flags

此字段指定4 7 > I自动重T激活D u件。|G一v位字段，" I 9C任何下P 4位 OR（或Y作）组合。

AP_RETRY_ON_START

在" T激活1，如果未从远L Z c S U =响&，则重T 4 7激活。如果在" T激活1，基>端Z G非活动D，则L序+ " T激活|。

AP_RETRY_ON_FAILURE

如果在活动和暂挂活动1 4 S ' \，则重T 4 7激活。如果在" T激活1，基>端Z发z故O，则L序+ " T激活|。

AP_RETRY_ON_DISCONNECT

如果4 7 I远L Z c以}#方=停止，则重T 4 7激活。

AP_DELAY_APPLICATION_RETRIES

I&C L序启动D 4 7激活重T（9C START_LS 或k s = 4 7激活）+ 9C **activation_delay_timer** x行w =。

AP_DELAY_INHERIT_RETRY

}K此字段中D j志指定D重T u件外，那些在基>端Z定义D **retry_flags** 字段中指定D重T u件也+ 9C。

ls_detail.def_data.max_activation_attempts

此字段；z z效果，}非在 **retry_flags** 中至Y h置一v j志。

C字段指定1远L Z c无响&或基>端Z；活动1，L序允许重T D次}。||括自动重T和I&C L序}动D激活" T = _。

如果达=C极限，则；再x行自动重T。C u件I STOP_LS、STOP_PORT、STOP_DLC 或一v I功激活4位。START_LS 或 OPEN_LU_SSCP_SEC_RQ z z一v %v激活" T，|在激活' \ 1；再重T。

c m >'无限制'。值 AP_USE_DEFAULTS <致在 DEFINE_PORT O a供D **max_activiation_attempts** D 9C。

ls_detail.def_data.activation_delay_timer

此字段；z z效果，}非在 **retry_flags** 中至Y h置一v j志。

C字段指定在自动重T之间L序H待D k }，以及如果在 **def_data.retry_flags** 中h置 AP_DELAY_APPLICATION_RETRIES 位，I&C L序}动D激活" T之间D k }。

值 AP_USE_DEFAULTS <致在 DEFINE_PORT O a供D **activiation_delay_timer** D 9C。

如果指定c，则L序+ 9C缺!计1器1间 30 k。

def_data.branch_link_type

ｖ BrNN。指定４７ＧＯ行４７或下行４７。ｖ１字段 **def_data.adj_cp_type** h置为 AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE 或 AP_BACK_LEVEL_LEN_NODE １此字段ＪＣ。

AP_UPLINK

Ｃ４７Ｇ一种Ｏ行４７。

AP_DOWNLINK

４７为下行４７。

如果字段 **adj_cp_type** h置为 AP_NETWORK_NODE，则此字段Ｘ须h置为 AP_UPLINK。

其｜Ｚｃ类型：此字段＋忽Ｔ。

ls_detail.det_data.adj_cp_is_brnn

ｖ BrNN。指定相Ｚ CP Ｇ否允许为、要ｓ为或｛止为一ｖ NN（BrNN）；例如，一ｖ BrNN 显＞其 NN Ｄ样子。ｖ１字段 **adj_cp_type** h置为 AP_NETWORK_NODE 或 AP_APPN_NODE（Ｒ在 XID ；换期间获知ＤＺｃ类型为网ｇＺｃ）１，此字段ＪＣ。

AP_BRNN_ALLOWED

相Ｚ CP 允许（＋；ＧＸ须）为一ｖ NN（BrNN）。

AP_BRNN_REQUIRED

相Ｚ CP ；允许为一ｖ NN（BrNN）。

AP_BRNN_PROHIBITED

相Ｚ CP ；允许为一ｖ NN（BrNN）。

如果字段 **adj_cp_type** h置为 AP_NETWORK_NODE，字段 **auto_act_supp** h置为 AP_YES，则此字段Ｘ须h置为 AP_BRNN_REQUIRED 或 AP_BRNN_PROHIBITED。

其｜Ｚｃ类型：此字段＋忽Ｔ。

如果因Ｎ｝错误而９动词；能执行，则Ｌ序返回下ＰＮ｝：

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LINK_NAME

AP_INVALID_LIST_OPTION

如果因Ｚｃ Ｐ未启动而９动词；能执行，则Ｌ序返回下ＰＮ｝：

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而９动词；能执行，则Ｌ序返回下ＰＮ｝：

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_LS_EXCEPTION

QUERY_LS 返回关Z在Z c定义D 4 7 > D信息P m。C信息I 『确定 }] 』（执行期间动，Q集D }] ）和『定义 }] 』（DEFINE_LS O D & C L序y a供D }] ）构I。

b些信息以 * 要q = 或详细信息 q = D P m返回。如需P关某v X定 LS D信息，或要获C几v 『段』中D P m信息， & h置 **ls_name** 字段。

否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽T C字段。 k N阅Z 12页D 『 i 询Z c 』，以获取关Z如何9 C P m q = D 3 0知6。

C P m y] **ls_name** 排序。W先4 { F $ 度排序，然后对相同 $ 度D { F x行 ASCII 词d ` 辑排序（ y] IBM D 6611 APPN MIB 定序）。如果选中K AP_LIST_FROM_NEXT，则返回D P m 4定义D次序（无 [指定D项G否存在）从下一项 * <。

要返回D 4 7 > D P m I 以 | G y t D端Z { F x行8选。在b种 i v下， & h置 **port_name** 字段（否则， C字段h置为全c）。

VCB a构

```
typedef struct query_ls_exception
{
        unsigned short  opcode;             /* verb operation code            */
        unsigned char   reserv2;            /* reserved                       */
        unsigned char   format;             /* format                         */
        unsigned short  primary_rc;         /* Primary return code            */
        unsigned long   secondary_rc;       /* Secondary return code          */
        unsigned char   *buf_ptr;           /* pointer to buffer              */
        unsigned long   buf_size;           /* buffer size                    */
        unsigned long   total_buf_size;     /* total buffer size required     */
        unsigned short  num_entries;        /* number of entries              */
        unsigned short  total_num_entries;  /* total number of entries        */
        unsigned char   list_options;       /* listing options                */
        unsigned char   reserv3;            /* reserved                       */
        unsigned long   exception_index;    /* index of LS exception entry    */
        unsigned char   ls_name;            /* name of link station           */
} QUERY_LS_EXCEPTION;

typedef struct LS_EXCEPTION
{
        unsigned short  overlay_size;       /* size of this entry             */
        unsigned long   exception_indes;    /* index of this entry            */
        unsigned_DATE_TIME
                        time;               /* date and time                  */
        unsigned char   ls_name[8];         /* link station name              */
        unsigned char   adj_cp_name[17];
                                            /* adjacent CP name               */
        unsigned char   adj_node_id[4];
                                            /* adjacent node id               */
        unsigned short  tg_number;          /* TG number                      */
        unsigned long   general_sense;      /* general sense data             */
        unsigned char   retry;              /* wil retry request              */
        unsigned long   end_sense;          /* termination sense data         */
        unsigned long   xid_local_sense;    /* XID local sense data           */
        unsigned long   xid_remote_sense;   /* XID remote sense data          */
        unsigned short  xid_error_byte;     /* offset of byte in error        */
        unsigned short  xid_error_bit;      /* offset of bit in error         */
        unsigned char   dlc_type;           /* DLC type                       */
```

```
        LINK_ADDRESS    local_addr;         /* local address                */
        LINK_ADDRESS    destination_addr;   /* destination address          */
        unsigned char   reserved[20];       /* reserved                     */
} LS_EXCEPTION;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_LS_EXCEPTION

format

j 6 VCB D q =。+此字段h置为 1，以指定O f P v D VCB D q = 1 f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4。

在下P N }中指定D **index** m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

exception_index

LS 异#项D w引。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

ls_name

返回相关项D 4 7 > { F。 | G在> X I显>字符集中D一v 8 字Z字符串。y P D 8 v字Z G P效D。如果C字段h置为U，则返回k任一4 7 >或全？4 7 >相关D项。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

QUERY_LS_EXCEPTION

buf_size

在缓ｅｘ中返回Ｄ信息＄度。

total_buf_size

返回值ｍｗ返回ｙＰ；ｋｓＤＰｍ信息ｙ需要Ｄ缓ｅｘＤ大小。Ｃ值Ｉ以大Ｚ **buf_size**。

num_entries

５际返回Ｄ项Ｄ｝？。

total_num_entries

已 - 返回Ｄ项Ｄ总｝。Ｃ值Ｉ以大Ｚ **num_entries**。

ls_exception.overlay_size

Ｃ项中Ｄ字Ｚ｝，即＝下一ｖ返回项（如果ＰＤ话）Ｄ偏移？。

ls_exception.exception_index

为Ｃ LS 异＃项分配Ｄｗ引。ｗ引Ｄ值从 0 ＊＜，在环绕之 0， ］增＝最大值 2**31-1（2,147,483,647）。

ls_exception.time

ｚＩ LS 异＃项Ｄ１间和日期。

ls_exception.ls_name

４７＞Ｄ｛字。｜Ｇ在＞ＸＩ显＞字符集中Ｄ一ｖ ８字Ｚ字符串。ｙＰＤ８ｖ字ＺＧＰ效Ｄ。

ls_exception.adj_cp_name

17 ｖ字ＺＤ全限定Ｚ｜ CP ｛Ｆ。｜Ｉ＝ｖＣ EBCDIC ｃ串＊Ｄ Ａ型 EBCDIC 字符串组Ｉ，ＲＩ EBCDIC ＵｑＲｎｄ。（？ｖ｛ＦＤ＄度最多Ｉ为８ ｖ字Ｚ，＂；含６入Ｕｑ。）Ｃ字段Ｄ值ｙ］下Ｐｉｖ确定:

如果以 XID 形＝ＳＵ＝Ｚ｜ CP ｛Ｆ，则返回Ｃ｛Ｆ。

如果以 XID 形＝ＳＵ＝Ｚ｜ CP ｛Ｆ，而一ｖ＞Ｘ定义Ｄ值ＩＣ，则返回Ｃ值。

否则，返回Ｕ。

ls_exception.node_id

在 XID ；换期间从Ｚ｜ＺｃＳＵＤＺｃ ID （如果；ＰＳＵ＝任何项，则为Ｕ）。ｂＧｖ ４字ＺＤ．ｙｘ制字符串。

ls_exception.tg_number

ｋ＝Ｃ４７＞Ｄ TG 关＊Ｄ号ｋ。范围从 0 ＝ 256。值 256 ｍ＞在＇＼１ TG 号未知。

ls_exception.general_sense

错误检ｂ｝］，｜ｋ４７激活Ｄ＊＜３序直＝ XID ３序Ｄ＊＜相关＊。ｂＩＺｃｚＩ。

ls_exception.retry

ｍ＞ＺｃＧ否重Ｔ启动ｋｓ以激活４７。

AP_NO

Ｚｃ；重Ｔ启动ｋｓ。

AP_YES

Ｚｃ重Ｔ启动ｋｓ。

ls_exception.end_sense
k 激活 " T D 终止相关 * D 检 b }] 。 b I DLC c z I 。

ls_exception.xid_local_sense
以 XID 形 = 发 M D > X z I D 检 b }] 。

ls_exception.xid.remote_sense
以 XID 形 = S U D 远 L z I D 检 b }] 。

ls_exception.xid_error_byte
XID 中错误字 Z 中错误位 D 偏移 ?（范围从 0 = 65535）。值 65535 m > C 字段无意义。

ls_exception.xid_error_bit
XID 中错误字 Z 中错误位 D 偏移 ?（范围从 0 = 7)。值 8 m > C 字段无意义。

ls_exception.dlc_type
DLC D 类型。 v 人通信或通信服务器支 V 下 P 类型:

AP_SDLC
AP_X25
AP_TR

= 加 D DLC 类型 I 以通过在 DEFINE_DLC 动词 O 指定新 D 类型来定义。 k N 阅 Z 49 页 D 『 DEFINE_DLC 』, 以获取 | 详细 D 信息。

ls_exception.local_addr.length
> X 4 7 > X 址 D $ 度。

ls_exception.local_address.address
> X 4 7 > D X 址。

ls_exception.destination_addr.length
Z | Z c O 目 D 4 7 > X 址 D $ 度。

ls_exception.destination_addr.address
Z | Z c O 目 D 4 7 > D X 址。

如果因 N } 错误而 9 动词；能执行, 则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_EXCEPTION_INDEX

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行, 则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行, 则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_LU_0_TO_3

QUERY_LU_0_TO_3 返回关Z 0、1、2 或 3 型> X LU D信息。C信息I『确定}]』（执行期间动，Q集D}]）和『定义}]』（DEFINE_LU_0_TO_3 OD&CL序ya供D}]）构I。

b些信息以*要q=或详细信息q=DPm返回。如需关Z某vX定>X LU D信息，或_获取几v『段』中DPm信息，&Ch置 **lu_name** 字段。否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽TC字段。

SNA API M户机Ovx支V某些N}。kN阅*记B*>以获取X定细Z。

C图jm>I能0响通信服务器和v人通信DY作D重要信息。

VCB a构

```
typedef struct query_lu_0_to_3
{
        unsigned short  opcode;            /* verb operation code           */
        unsigned char   attributes;        /* Verb attributes               */
        unsigned char   reserv2;           /* reserved                      */
        unsigned char   format;            /* format                        */
        unsigned short  primary_rc;        /* primary return code           */
        unsigned long   secondary_rc;      /* secondary return code         */
        unsigned char   *buf_ptr;          /* pointer to buffer             */
        unsigned long   buf_size;          /* buffer size                   */
        unsigned long   total_buf_size;    /* total buffer size required    */
        unsigned short  num_entries;       /* number of entries             */
        unsigned short  total_num_entries; /* total number of entries       */
        unsigned char   list_options;      /* listing options               */
        unsigned char   reserv3;           /* reserved                      */
        unsigned char   pu_name[8];        /* PU name filter                */
        unsigned char   lu_name[8];        /* LU name                       */
        unsigned char   host_attachment;   /* Host attachment filter        */
} QUERY_LU_0_TO_3;

typedef struct lu_0_to_3_summary
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   pu_name[8];        /* PU name                       */
        unsigned char   lu_name[8];        /* LU name                       */
        unsigned char   description[RD_LEN];
                                           /* resource description          */
        unsigned char   nau_address;       /* NAU address                   */
        unsigned char   lu_sscp_sess_active;
                                           /* Is LU-SSCP session active     */
        unsigned char   appl_conn_active;  /* Is connection to appl active? */
        unsigned char   plu_sess_active;   /* Is PLU-SLU session active     */
        unsigned char   host_attachment;   /* LU's host attachment          */
} LU_0_TO_3_SUMMARY;

typedef struct lu_0_to_3_detail
{
        unsigned short  overlay_size;      /* size of this entry            */
        unsigned char   lu_name[8];        /* LU name                       */
        unsigned char   reserv1[2];        /* reserved                      */
        LU_0_TO_3_DET_DATA det_data;       /* Determined data               */
        LU_0_TO_3_DEF_DATA def_data;       /* Defined data                  */
} LU_0_TO_3_DETAIL;
```

```
typedef struct lu_0_to_3_det_data
{
        unsigned char   lu_sscp_sess_active;
                                        /* Is LU-SSCP session active     */
        unsigned char   appl_conn_active;  /* Application is using LU    */
        unsigned char   plu_sess_active;   /* Is PLU-SLU session active  */
        unsigned char   host_attachment;   /* Host attachment            */
        SESSION_STATS   lu_sscp_stats;     /* LU-SSCP session statistics */
        SESSION_STATS   plu_stats;         /* PLU-SLU session statistics */
        unsigned char   plu_name[8];       /* PLU name                   */
        unsigned char   session_id[8];     /* Internal ID of PLU-SLU sess */
        unsigned char   app_spec_det_data[256];
                                        /* Application Specified Data    */
        unsigned char   app_type;          /* Application type           */
        unsigned char   sscp_id[6];        /* SSCP ID                    */
        unsigned char   bind_lu_type;      /* LU type issuing BIND       */
        unsigned char   reserva[12];       /* reserved                   */
} LU_0_TO_3_DET_DATA;

typedef struct session_stats
{
        unsigned short  rcv_ru_size;       /* session receive RU size    */
        unsigned short  send_ru_size;      /* session send RU size       */
        unsigned short  max_send_btu_size; /* max send BTU size          */
        unsigned short  max_rcv_btu_size;  /* max rcv BTU size           */
        unsigned short  max_send_pac_win;  /* max send pacing win size   */
        unsigned short  cur_send_pac_win;  /* current send pacing win size */
        unsigned short  max_rcv_pac_win;   /* max receive pacing win size */
        unsigned short  cur_rcv_pac_win;   /* current receive pacing     */
                                        /* window size                   */
        unsigned long   send_data_frames;  /* number of data frames sent */
        unsigned long   send_fmd_data_frames;
                                        /* num of FMD data frames sent   */
        unsigned long   send_data_bytes;   /* number of data bytes sent  */
        unsigned long   rcv_data_frames;   /* num data frames received   */
        unsigned long   rcv_fmd_data_frames;
                                        /* num of FMD data frames recvd  */
        unsigned long   rcv_data_bytes;    /* number of data bytes received */
        unsigned char   sidh;              /* session ID high byte       */
        unsigned char   sidl;              /* session ID low byte        */
        unsigned char   odai;              /* ODAI bit set               */
        unsigned char   ls_name[8];        /* Link station name          */
        unsigned char   pacing_type;       /* type of pacing in use      */
} SESSION_STATS;

typedef struct lu_0_to_3_def_data
{
        unsigned char   description[RD_LEN];
                                        /* resource description          */
        unsigned char   nau_address;       /* LU NAU address             */
        unsigned char   pool_name[8];      /* LU Pool name               */
        unsigned char   pu_name[8];        /* PU name                    */
        unsigned char   priority;          /* LU priority                */
        unsigned char   lu_model;          /* LU model                   */
        unsigned char   sscp_id[6];
        unsigned char   timeout;           /* Timeout                    */
        unsigned char   app_spec_def_data[16];
                                        /* Application Specified Data    */
        unsigned char   model_name[7];     /* LU model                   */
        unsigned char   reserv3[17];         /* reserved                 */
} LU_0_TO_3_DEF_DATA;
```

提供N数

&C L序a供下P N }:

QUERY_LU_0_TO_3

opcode

AP_QUERY_LU_0_TO_3

attributes

动词D t 性。C字段G一 v 位字段。Z一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段 h 置为 c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z 写入P m信息D缓 e x D指 k。

buf_size

y a 供D缓 e x D大小。返回D }] + ；，过 b v 大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过 b v 值。 c 值m > 无限制。

list_options

m w在P m信息中& C返回 2 4。

AP_SUMMARY

v 返回 * 要信息。

对Z SNA API M户机还支V AP_SUMMARY 值。

AP_DETAIL

返回详细信息。

指定D **lu_name**（见下P N }）m > w引值， C w引值C Z 指定要返回D 5 际信息D起 < c。

AP_FIRST_IN_LIST

忽T **session_id**，返回 P m从P m D Z 一项 * <。

对Z SNA API M户机还支V AP_FIRST_IN_LIST 值。

.

AP_LIST_FROM_NEXT

返回 P m从P m中 4 a 供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回 P m从w引值 y 指定D那项 * <。

lu_name

} i 询D > X LU D { F。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽T C字段。

lu_0_to_3_summary.pu_name 值未返回至 SNA API M户机。

lu_0_to_3_summary.lu_name

} i 询D > X LU D { F。 b G一v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一v字母 * 头）， " 以 EBCDIC U q R n d。

lu_0_to_3_summary.description

资源 5 w（如在 DEFINE_LU_0_TO_3 中 5 w D）。 | G一v以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

lu_0_to_3_summary.description 值未返回至 SNA API M户机。

lu_0_to_3_summary.nau_address

LU D网 g I 寻址%元X址，范围在 1--255 内。

lu_0_to_3_summary.nau_address 值未返回至 SNA API M户机。

lu_0_to_3_summary.lu_sscp_sess_active

5 w LU-SSCP 会话G否G活动D (AP_YES 或 AP_NO)。

lu_0_to_3_summary.lu_sscp_sess_active 值未返回至 SNA API M户机。

lu_0_to_3_summary.appl_conn_active

指定& C L 序G否 9 C C LU（AP_YES 或 AP_NO）。

lu_0_to_3_summary.aapl_conn_active 值未返回至 SNA API M户机。

lu_0_to_3_summary.plu_sess_active

指定 PLU-SLU 会话G否G活动D（AP_YES 或 AP_NO）。

lu_0_to_3_summary.plu_sess_active 值未返回至 SNA API M户机。

lu_0_to_3_summary.host_attachment

LU 主机， S 类型:

AP_DLUR_ATTACHED

LU 9 C DLUR 同主机系统相，。

AP_DIRECT_ATTACHED

LU 同主机系统直S ， S。

lu_0_to_3_detail.overlay_size

C项中D字Z }， 即 = 下一v返回项（如果P D话）D偏移？。

lu_0_to_3_detail.lu_name
} i 询D > X LU D { F。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。

lu_0_to_3_detail.det_data.lu_sscp_sess_active
5 w LU-SSCP 会话G否G活动D (AP_YES 或 AP_NO)。

lu_0_to_3_detail.det_data.appl_conn_active
指定C LU 1 0 G否 } I 一 v & C L 序在 9 C（AP_YES 或 AP_NO）。

lu_0_to_3_detail.det_data.plu_sess_active
指定 PLU-SLU 会话G否G活动D（AP_YES 或 AP_NO）。

lu_0_to_3_detail.det_data.host_attachment
LU 主机， S 类型:

AP_DLUR_ATTACHED
LU 9 C DLUR 同主机系统相，。

AP_DIRECT_ATTACHED
LU 同主机系统直S ， S。

lu_0_to_3_detail.det_data.lu_sscp_stats.rcv_ru_size
< 终 # t C字段。

lu_0_to_3_detail.det_data.lu_sscp_stats.send_ru_size
< 终 # t C字段。

lu_0_to_3_detail.det_data.lu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

lu_0_to_3_detail.det_data.lu_sscp_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

lu_0_to_3_detail.det_data.lu_sscp_stats.max_send_pac_win
< 终 + C字段 h 置为 c 。

lu_0_to_3_detail.det_data.lu_sscp_stats.cur_send_pac_win
< 终 + C字段 h 置为 c 。

lu_0_to_3_detail.det_data.lu_sscp_stats.max_rcv_pac_win
< 终 + C字段 h 置为 c 。

lu_0_to_3_detail.det_data.lu_sscp_stats.cur_rcv_pac_win
< 终 + C字段 h 置为 c 。

lu_0_to_3_detail.det_data.lu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ?。

lu_0_to_3_detail.det_data.lu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

lu_0_to_3_detail.det_data.lu_sscp_stats.send_data_bytes
发M D } # }] w D字Z }。

lu _0_to_3_detail.det_data.lu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

lu_0_to_3_detail.det_data.lu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

lu_0_to_3_detail.det_data.lu_sscp_stats.rcv_data_bytes
S U D } # }] w D字 Z } 。

lu_0_to_3_detail.det_data.lu_sscp_stats.sidh
会话 ID _ 字 Z 。

lu_0_to_3_detail.det_data.lu_sscp_stats.sidl
会话 ID M字 Z 。

lu_0_to_3_detail.det_data.lu_sscp_stats.odai
起 < 目 D X 址指 > 符。 1 (" 会话 1 , 如果 > X Z c | 含主 4 7 > , 则 ACTLU D发M方置C字段为0; " R如果 ACTLU D发M方G | 含从 4 7 > D Z c , 则置C字段为 1。

lu_0_to_3_detail.det_data.lu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P效D。 C字段能C Z Q会话D统计信息通过信息w ? 同 4 7 关 * 起来。

lu_0_to_3_detail.det_data.lu_sscp_stats.pacing_type
LU-SSCP 会话中 } 在 9 C D S U w = 类型。 + h 置为 AP_NONE。

lu_0_to_3_detail.det_data.plu_stats.rcv_ru_size
最大 S U RU 大小。

lu_0_to_3_detail.det_data.plu_stats.send_ru_size
最大发M RU 大小。

lu_0_to_3_detail.det_data.plu_stats.max_send_btu_size
能发M D最大 BTU 大小。

lu_0_to_3_detail.det_data.plu_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

lu_0_to_3_detail.det_data.plu_stats.max_send_pac_win
会话中发M w = 窗 Z D最大 _ 寸。

lu_0_to_3_detail.det_data.plu_stats.cur_send_pac_win
会话中发M w = 窗 Z D 1 0 大小。

lu_0_to_3_detail.det_data.plu_stats.max_rcv_pac_win
会话中 S U w = 窗 Z D最大 _ 寸。

lu_0_to_3_detail.det_data.plu_stats.cur_rcv_pac_win
会话中 S U w = 窗 Z D 1 0 大小。

lu_0_to_3_detail.det_data.plu_stats.send_data_frames
发M D } # w }] 帧D } ? 。

lu_0_to_3_detail.det_data.plu_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ? 。

lu_0_to_3_detail.det_data.plu_stats.send_data_bytes
发M D } # }] w D字 Z } 。

lu_0_to_3_detail.det_data.plu_stats.rcv_data_frames
S U D } # w }] 帧D } ? 。

lu_0_to_3_detail.det_data.plu_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ? 。

lu_0_to_3_detail.det_data.plu_stats.rcv_data_bytes
S U D } # }] w D字 Z }。

lu_0_to_3_detail.det_data.plu_stats.sidh
会话 ID _字 Z。

lu_0_to_3_detail.det_data.plu_stats.sidl
会话 ID M字 Z。

lu_0_to_3_detail.det_data.plu_stats.odai
起 < 目 D X址指 > 符。 * < 会话 1，如果 > X Z c | 含主 4 7 >，则 BIND D 发M方 + C字段 h 置为 0; 而如果 BIND 发M方 G | 含次级 4 7 > D Z c，则 Q C字段 h 置为 1。

lu_0_to_3_detail.det_data.plu_stats.ls_name
同统计信息相关 * D 4 7 > { F。 | G在 > X I 显 > 字符集中 D一 v 8 字 Z字符串。 y P D 8 v字 Z G P效 D。

lu_0_to_3_detail.det_data.plu_stats.pacing_type
PLU-SSCP 会话中 } 在 9 C D S U w = 类型。 C N } 能取 AP_NONE 或 AP_PACING_FIXED 值。

lu_0_to_3_detail.det_data.plu_name
主 LU { F。 b G一 v 8 字 Z字母 } 字 D A 型 EBCDIC 字符串（以一 v字母 * 头），" 以 EBCDIC U q R n d。（如果 PLU-SLU G非活动 D，则 # t C字段）。

lu_0_to_3_detail.det_data.session_id
PLU-SLU 会话 D 8 v字 Z D内？ j 6。

lu_0_to_3_detail.det_data.app_spec_det_data
t。

lu_0_to_3_detail.det_data.sscp_id
b G v | 含从 C LU 9 C PU D ACTPU 中 S U D SSCP ID D 6 字 Z D字段。

如果 **lu_sscp_sess_active** ；G AP_YES，则 C字段 h 置为 c。

lu_0_to_3_detail.det_data.app_type
t。

lu_0_to_3_detail.lu_0_to_3_det_data.bind_lu_type
发 v 原 < BIND D LU D LU 类型。如果 P 活动 D LU-LU 会话，则 I 能 G 下 P 之一:

AP_LU_TYPE_0
AP_LU_TYPE_1
AP_LU_TYPE_2
AP_LU_TYPE_3
AP_LU_TYPE_6 （对 Z 下 N从 t LU 6.2）

如果；P 活动 D LU--LU 会话，则 C字段取下 P 值:

AP_LU_TYPE_UNKNOWN

lu_0_to_3_detail.def_data.description
资源 5 w（如在 DEFINE_LU_0_TO_3 中 5 w D）。 | G 一 v 以 > X I 显 > 字符集 m > D 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

lu_0_to_3_detail.def_data.nau_address
LU D网 g I 寻址 % 元 X 址，范围在 1--255 内。

lu_0_to_3_detail.def_data.pool_name
C LU y t D X D { F。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。如果 LU ； t Z 某一 X，则此字段 + ； h 置为全二 x 制 c。

lu_0_to_3_detail.def_data.pu_name
此 LU + 9 C D PU（在 DEFINE_LS 动词 O 指定）D { F。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d。

lu_0_to_3_detail.def_data.priority
在发 M = 主机 1 D LU E 先级。 | h 置为下 P 值之一:

AP_NETWORK
AP_HIGH
AP_MEDIUM
AP_LOW

lu_0_to_3_detail.def_data.lu_model
LU D模型类型和号 k。 | h 置为下 P 值之一:

AP_3270_DISPLAY_MODEL_2
AP_3270_DISPLAY_MODEL_3
AP_3270_DISPLAY_MODEL_4
AP_3270_DISPLAY_MODEL_5
AP_RJE_WKSTN
AP_PRINTER
AP_SCS_PRINTER
AP_UNKNOWN

lu_0_to_3_detail.def_data.sscp_id
此字段指定允许激活此 LU D SSCP D ID。 | G 一 v 6 字 Z 二 x 制字段。如果字段 h 置为二 x 制 c，则 LU 也许 I I 任何 SSCP 激活。

lu_0_to_3_detail.def_data.timeout
LU D指定，1 值（ k ）。如果 a 供 K 某一，1 值， " R LU D C 户在 OPEN_LU_SSCP_SEC_RQ O 指定 K **allow_timeout** （或 _，如为 PU 集中，则在下 N LU O 定义），则如在 C 指定 1 间内 PLU-SLU 会话 # V 非活动， R 存在下 P u 件之一，则 LU + ；M 放:

- 会话在 \ 限资源 4 7 O 传]
- m 一 & C 要在此会话再次 9 C 0 9 C LU

如果，1 h 置为 c，则 LU ；会 M 放。

lu_0_to_3_detail.def_data.app_spec_def_data
来自 DEFINE_LU_0_TO_3 D & C L 序指定 D }]； L 序； b M C 字段， | 只 G 简 % 存储 " 在 QUERY_LU_0_TO_3 动词 O 返回。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_INVALID_LIST_OPTION

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_LU_POOL

QUERY_LU_POOL 返回X D P m， " R LU t Z b些X。

b些信息以 * 要q = 或详细信息q = D P m返回。如需P关某v X定 LU X D信息，或要获C几v『段』中D P m信息，& h置 **pool_name** 和 **lu_name** 字段。如果 **lu_name** 字段h置为全c，则返回D信息从指定X D Z一v LU * <。如果 **list_options** 字段h置为 AP_FIRST_IN_LIST，则忽T b些字段中D某一v字段。

VCB a构

```
typedef struct query_lu_pool
{
        unsigned short  opcode;              /* verb operation code         */
        unsigned char   attributes;          /* verb attributes             */
        unsigned char   format;              /* format                      */
        unsigned short  primary_rc;          /* primary return code         */
        unsigned long   secondary_rc;        /* secondary return code       */
        unsigned char   *buf_ptr;            /* pointer to buffer           */
        unsigned long   buf_size;            /* buffer size                 */
        unsigned long   total_buf_size;      /* total buffer size required  */
        unsigned short  num_entries;         /* number of entries           */
        unsigned short  total_num_entries;   /* total number of entries     */
        unsigned char   list_options;        /* listing options             */
        unsigned char   reserv3;             /* reserved                    */
        unsigned char   pool_name[8];        /* pool name                   */
        unsigned char   lu_name[8];          /* LU name                     */
} QUERY_LU_POOL;

typedef struct lu_pool_summary
{
        unsigned short  overlay_size;        /* size of this entry          */
        unsigned char   pool_name[8];        /* pool name                   */
        unsigned char   description[RD_LEN]; /* resource description        */
        unsigned short  num_active_lus;      /* num of currently active LUs */
        unsigned char   num_avail_lus;       /* num of currently available  */
                                             /* LUs                         */
} LU_POOL_SUMMARY;

typedef struct lu_pool_detail
{
        unsigned short  overlay_size;        /* size of this entry          */
        unsigned char   pool_name[8];        /* pool name                   */
        unsigned char   description[RD_LEN]; /* resource description        */
        unsigned char   lu_name[8];          /* LU name                     */
        unsigned char   lu_sscp_sess_active; /* Is LU-SSCP session active   */
        unsigned char   appl_conn_active;    /* Is SSCP connection open     */
        unsigned char   plu_sess_active;     /* Is PLU-SLU session active   */
} LU_POOL_DETAIL;
```

提供N数

& C L序a供下P N }:

opcode

AP_QUERY_LU_POOL

attributes

动词D t性。C字段G一v位字段。Z一位 | 含要定义资源D I见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format
j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

buf_ptr
指向C Z 写入 P m信息D缓 e x D指 k 。

buf_size
y a 供D缓 e x D大小。返回D }] + ； ，过 b v 大小。

num_entries
要返回D项D最大 } ? 。项D } ? ；要，过 b v 值。 c 值m > 无限制。

list_options
m w在 P m信息中& C返回 2 4:

AP_SUMMARY
v 返回 * 要信息。

AP_DETAIL
返回详细信息。

指定D **pool_name** 和 **lu_name** D组合（见下 P N } ）m > w引值，C w引值C Z 指定要返回D 5 际信息D起 < c 。

AP_FIRST_IN_LIST
忽T w引值，返回 P m从 P m D Z 一项 * < 。

AP_LIST_FROM_NEXT
返回 P m从 P m中 4 a 供D w引值 y 指定D那项D下一项 * < 。

AP_LIST_INCLUSIVE
返回 P m从w引值 y 指定D那项 * < 。

pool_name
LU X D { F 。此 { F G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽T C字段。

lu_name
LU { F 。此 { F G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。如果 h 置为全二 x 制 c， t Z 指定X D LU 从X D * < 起 P v 。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

buf_size
在缓 e x 中返回 D 信息 $ 度。

total_buf_size
返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。C值 I 以大Z **buf_size**。

num_entries
返回D目 < 项D } 目。

total_num_entries
已 - 返回D项D总 } 。C值 I 以大Z **num_entries**。

lu_pool_summary.overlay_size
C项中D字Z }，即 = 下一 v 返回项（如果 P D话）D偏移 ? 。

lu_pool_summary.pool_name
指定 LU y t LU X D { F。b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头），"以 EBCDIC U q R n d。（注意，如果C字段 4 k s 指定，" R **lu_name** 字段 h 置为全二 x 制 c，则 v 返回X中D LU。）

lu_pool_summary.description
LU X h v（如在 DEFINE_LU_POOL 中 5 w D）。

lu_pool_summary.num_active_lus
在 5 P 活动D LU-SSCP 会话D指定X中D LU } ? 。

lu_pool_summary.num_avail_lus
指定X中 ? 分 LU D } ?，b些 LU I 以 z 足 **open_force** h 置为 AP_YES D OPEN_LU_SSCP_SEC_REQ。| | 括 y P 符合下P u 件D LU: | D PU G 活动D，或 _ | D主机 4 7 G自动激活D，" R | D，S G自 I D。b v } 目；管 LU **model_type**、 **model_name** 和 PU D DDDLU 支V。b I 能H z 足 OPEN_LU_SSCP_SEC_REQ （为 **model_type** 指定X b 值）D LU Y。

lu_pool_detail.num_active_lus
在 5 P 活动D LU-SSCP 会话D指定X中D LU } ? 。

lu_pool_detail.num_avail_lus
在 5 P I C D LU-SSCP 会话D指定X中D LU } ? 。

lu_pool_detail.overlay_size
C项中D字Z }，即 = 下一 v 返回项（如果 P D话）D偏移 ? 。

lu_pool_detail.pool_name
指定 LU y t LU X D { F。b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头），"以 EBCDIC U q R n d。（注意，如果C字段 4 k s 指定，" R **lu_name** 字段 h 置为全二 x 制 c，则 v 返回X中D LU。）

lu_pool_detail.description
LU h v（如在 DEFINE_LU_0_TO_3 中 5 w D）。

lu_pool_detail.lu_name
t Z X D LU D LU { F。b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头），"以 EBCDIC U q R n d。如果C { F h 置为全 c，那 4 b m > 指定X G U D。

lu_pool_detail.lu_sscp_sess_active
5 w LU-SSCP 会话G否G活动D (AP_YES 或 AP_NO)。

lu_pool_detail.appl_conn_active
指定 LU 会话 1 0 G否 } I 一 v & C L 序在 9 C（AP_YES 或 AP_NO）。

lu_pool_detail.plu_sess_active
指定 PLU-SLU 会话G否G活动D（AP_YES 或 AP_NO）。

如果因N｝错误而 9 动词；能执行，则L序返回下P N｝:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LIST_OPTION

AP_INVALID_POOL_NAME
AP_INVALID_LU_NAME

如果因Z c P未启动而 9 动词；能执行，则L序返回下P N｝:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L序返回下P N｝:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_MDS_APPLICATION

QUERY_MDS_APPLICATION 返回& C L 序D P m， b些& C L 序已 - 注 a K MDS 级消息。

发 v Z 603页D 『 Z 15 B 管理服务动词』”。〈 ” ≡ ℃ ≡ £ ’ Σ 《 ” “ ≡ ∧ 〃 』” 〈 ” ≡ ℃ ≥ £

如 侗洞 侗 对舵洞冻 侗 对舵侗懂栋陡恫 侗 栋 对赌栋懂栋

application V 员源 **list_options** 操绦云远 Z12页D 『 询Z c 』以获准 e 梗 a 辉 t 台 m 侗 化 黄 e 换 T 迁 0 辉 苗 栋

unsign

} QUERY_MD

typedef st

unsign

unsign

} 录MDS _ APPL

list_options

b m w P m信息中要返回 2 4: 指定D **application**（见下P N }）m > w引值, C w引值C Z指定要返回D 5际信息D起 < c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

application

& C L序{。{ F G v 8 字Z字母}字D A 型 EBCDIC 字符串。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size

返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总}。C值I以大Z **num_entries**。

mds_application_data.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

mds_application_data.application

注a & C L序D{ F。{ F G v 8 字Z字母}字D A 型 EBCDIC 字符串。

mds_application_data.max_rcv_size

& C L序在一段中I以S U D最大字Z }（1 & C L序注a MDS 1已指定)。如需关Z MDS-级& C L序注a D | 详细信息, k N < Z 15B 管理服务动词。

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_APPLICATION_NAME

AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_MDS_STATISTICS

QUERY_MDS_STATISTICS 返回管理服务统计信息。C动词 I C Z 估计 MDS 7 I 选择通信？D级 p 。

VCB a 构

```
typedef struct query_mds_statistics
{
        unsigned short  opcode;             /* verb operation code         */
        unsigned char   reserv2;            /* reserved                    */
        unsigned char   format;             /* format                      */
        unsigned short  primary_rc;         /* primary return code         */
        unsigned long   secondary_rc;       /* secondary return code       */
        unsigned long   alerts_sent;        /* number of alert sends       */
        unsigned long   alert_errors_rcvd;  /* error messages received     */
                                            /* for alert sends             */
        unsigned long   uncorrelated_alert_errors;
                                            /* uncorrelated alert          */
                                            /* errors received             */
        unsigned long   mds_mus_rcvd_local; /* number of MDS_MUs received  */
                                            /* from local applications     */
        unsigned long   mds_mus_rcvd_remote;
                                            /* number of MDS_MUs received  */
                                            /* from remote applications    */
        unsigned long   mds_mus_delivered_local;
                                            /* num of MDS_MUs delivered    */
                                            /* to local applications       */
        unsigned long   mds_mus_delivered_remote;
                                            /* num of MDS_MUs              */
                                            /* delivered to remote appls   */
        unsigned long   parse_errors;       /* number of MDS_MUs received  */
                                            /* with parse errors           */
        unsigned long   failed_deliveries;  /* number of MDS_MUs where     */
                                            /* delivery failed             */
        unsigned long   ds_searches_performed;
                                            /* number of DS searches done  */
        unsigned long   unverified_errors;  /* number of unverified errors */
        unsigned char   reserva[20];        /* reserved                    */
} QUERY_MDS_STATISTICS;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_MDS_STATISTICS

format

j 6 VCB D q = 。 + C字段 h 置为 c , 以指定O f P v D VCB D f > 。

返回N数

如果动词 I 功执行, 则L 序返回下 P N }:

primary_rc

AP_OK

alerts_sent

9 C MDS 传M系统发M D > X z I D / (} 。

QUERY_MDS_STATISTICS

alert_errors_rcvd

I MDS S U D m w因信息 | 含 / (而 < 致传] ' \ D错误信息 } 。

uncorrelated_errors_rcvd

I MDS S U D m w因信息 | 含 / (而 < 致传] ' \ D错误信息 } 。在错误信息无法 k MDS 发M D / (队 P O D / ((" 关系 1 , 发 z 传] ' \ 。 MDS 维护固定大小D队 P , C队 P 中 _ Y 缓存发M至问 b 确定 9 c D / (。一) 队 P z K , + 废弃最 I D / (, I 新D来代 f 。如果 S U = 传] 错误信息, MDS " T + 错误信息 k _ Y 缓存D / ((" 关系, 9 / (I 以一直 # V = 问 b 确定 9 c 4 原后。

" : + 维护 b = v } 值 **alert_errors_rcvd** 和 **uncorrelated_errors_rcvd**, 以 9 发M / (队 P D大小 I 以 w { 。**uncorrelated_errors_rcvd** D] 增值 , 次 m w发M / (队 P + 小。

mds_mus_rcvd_local

从 > X & C L 序 S U = D MDS_MU D } ? 。

mds_mus_rcvd_remote

9 C MDS_RECEIVE 和 MSU_HANDLER B 务 L 序从远 L Z c S U = D MDS_MU D } ? 。

mds_mus_delivered_local

I 功传] 至 > X & C L 序D MDS_MU } ? 。

mds_mus_delivered_remote

9 C MDS_SEND B 务 L 序 I 功传] 至远 L Z c D MDS_MU } ? 。

parse_errors

S U = D | 含头 q = 错误D MDS_MU } ? 。

failed_deliveries

C Z c 传] ' \ D MDS_MU } ? 。

ds_searches_performed

t 。

unverified_errors

t 。

如果因 Z c P 未启动而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_MODE

QUERY_MODE 返回关Z > X LU 及X b伙i LU 已9 C D方= D返回。 b些信息以*要q =或详细信息q = D P m返回。如需P关某v X定方= D信息，或要获C几v『段』中D P m信息，& h置 **mode_name** 字段。否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽T C字段。注意: X须h置 **lu_name**（或 **lu_alias**）和 **plu_alias**（或 **fqplu_name**）字段。如果C字段非c，则 **lu_name** D 9 C + E先Z **lu_alias**。 k N阅Z 12页D『i询Z c』，以获取关Z如何9 C P m q = D 3 0知6。

P m v |括I **lu_name**（或 **lu_alias**）指定D > X LU D信息。C P m D次序G **fqplu_name** 后z **mode_name**。 W先4 { F $度排序，然后对相同$度D { F x行ASCII 词d ` 辑排序（y] j准 MIB 定序）。

如果 **plu_alias** h置为全c，则+ 9 C **fqplu_name** 值，否则 **plu_alias** +一直；9 C " R忽T **fqplu_name**。

返回D方= D P m I y] | G 1 0 G否P任何活动会话来x行8选。如果希望8选，则& C + **active_sessions** 字段h置为 AP_YES （否则，& + C字段h置为AP_NO）。C动词返回在> X LU 和伙i LU * < 9 C方= 1 M确定D信息。QUERY_MODE_DEFINITION 动词v返回定义信息.

VCB a构

```
typedef struct query_mode
{
        unsigned short  opcode;             /* verb operation code          */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   format;             /* format                       */
        unsigned short  primary_rc;         /* primary return code          */
        unsigned long   secondary_rc;       /* secondary return code        */
        unsigned char   *buf_ptr;           /* pointer to buffer            */
        unsigned long   buf_size;           /* buffer size                  */
        unsigned long   total_buf_size;     /* total buffer size required   */
        unsigned short  num_entries;        /* number of entries            */
        unsigned short  total_num_entries;  /* total number of entries      */
        unsigned char   list_options;       /* listing options              */
        unsigned char   reserv3;            /* reserved                     */
        unsigned char   lu_name[8];         /* LU name                      */
        unsigned char   lu_alias[8];        /* LU alias                     */
        unsigned char   plu_alias[8];       /* partner LU alias             */
        unsigned char   fqplu_name[17];     /* fully qualified partner      */
                                            /* LU name                      */
        unsigned char   mode_name[8];       /* mode name                    */
        unsigned char   active_sessions;    /* active sessions only filter  */
} QUERY_MODE;

typedef struct mode_summary
{
        unsigned short  overlay_size;       /* size of this entry           */
        unsigned char   mode_name[8];       /* mode name                    */
        unsigned char   description[RD_LEN];
                                            /* resource description         */
        unsigned short  sess_limit;         /* current session limit        */
        unsigned short  act_sess_count;     /* curr active sessions count   */
        unsigned char   fqplu_name[17];     /* partner LU name              */
        unsigned char   reserv1[3];         /* reserved                     */
} MODE_SUMMARY;
```

QUERY_MODE

```
typedef struct mode_detail
{
        unsigned short  overlay_size;       /* size of this entry           */
        unsigned char   mode_name[8];       /* mode name                    */
        unsigned char   description[RD_LEN];
                                            /* resource description         */
        unsigned short  sess_limit;         /* session limit                */
        unsigned short  act_sess_count;     /* currently active sess count  */
        unsigned char   fqplu_name[17];     /* partner LU name              */
        unsigned char   reserv1[3];         /* reserved                     */
        unsigned short  min_conwinners_source;
                                            /* min conwinner sess limit     */
        unsigned short  min_conwinners_target;
                                            /* min conloser limit           */
        unsigned char   drain_source;       /* drain source?                */
        unsigned char   drain_partner;      /* drain partner?               */
        unsigned short  auto_act;           /* auto activated conwinner     */
                                            /* session limit                */
        unsigned short  act_cw_count;       /* active conwinner sess count  */
        unsigned short  act_cl_count;       /* active conloser sess count   */
        unsigned char   sync_level;         /* synchronization level        */
        unsigned char   default_ru_size;    /* default RU size to maximize  */
                                            /* performance                  */
        unsigned short  max_neg_sess_limit; /* max negotiated session limit */
        unsigned short  max_rcv_ru_size;    /* max receive RU size          */
        unsigned short  pending_session_count;
                                            /* pending sess count for mode  */
        unsigned short  termination_count;  /* termination count for mode   */
        unsigned char   implicit;           /* implicit or explicit entry   */
        unsigned char   reserva[15];        /* reserved                     */
} MODE_DETAIL;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_MODE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **lu_name**（或_， 如果 **lu_name** h置为全c，则为 **lu_alias**）、**plu_alias**（或_， 如果 **plu_alias** h置为全c，则为

fqplu_name）、 **mode_name**（见下P N }）m > w引值， C w引值 C Z指定要返回D 5际信息D起< c。 1指定伙 i LU w引1，如果I 能D话，在P m中 | 括关Z其 | 伙 i LU D信息。

AP_FIRST_IN_LIST
如果 **plu_alias** 和 **fqplu_name** h置为全c，则返回P m从P m D Z 一 v伙 i LU * <， " R忽T **mode_name** w引。如果指定K **plu_alias** 或 **fqplu_name**，则P m在C w引处* <， + **mode_name** w引值；忽T， " R返回P m从P m D Z一 v方 = 项* <。

AP_LIST_FROM_NEXT
返回P m从P m中4 a供D w引值 y 指定D那项D下一项* <。

AP_LIST_INCLUSIVE
返回P m从w引值 y 指定D那项* <。

lu_name
LU { F。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段 h 置为全 c，则 **lu_alias** 字段 + C Z确定w引。

lu_alias
> X定义D LU p {。 | G在> X I 显> 字符集中D一 v 8 字Z字符串。 v 1 **lu_name** 字段 h 置为全 c 1， | E P效，在 b 种 i v下。 y P 8 v字Z都G P效D " R X须 h 置。如果 **lu_name** 和 **lu_alias** 字段都 h 置为全 c，则 9 C k X制 c（缺！ LU）相关* D LU。

plu_alias
对方 LU D p {。 | G在> X I 显> 字符集中D一 v 8 字Z字符串。 y P 8 v 字Z都G P效D， " R X须 h 置。如果C字段 h 置为全 c，则 **fqplu_name** 字段 + C Z确定w引。

fqplu_name
伙 i LU D 17 字Z全限定网 g { F。C { F G I = v C EBCDIC c 串* D A 型 EBCDIC 字符串构I， " C EBCDIC U q R n d。（？v { F D $ 度最多I为 8 v字Z， " ；含 6 入U q。）

mode_name
为一组会话指定网 g t 性D方 = {。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母* 头）， " 以 EBCDIC U q R n d。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽T C字段。

active_sessions
活动会话过K器。指定返回方 = G否要 y] | G 1 0 P无活动对话来 x 行 8 选（AP_YES 或 AP_NO）。

返回N数

如果动词I功执行，则L序返回下P N }：

primary_rc
AP_OK

buf_size
在缓 e x 中返回D信息 $ 度。

total_buf_size
返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。C值 I 以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值 I 以大Z **num_entries**。

mode_summary.overlay_size
C项中D字Z }, 即 = 下一 v 返回项（如果 P D话）D偏移 ?。

mode_summary.mode_name
为一组会话指定网 g t 性D方 = { 。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）, " 以 EBCDIC U q R n d。

mode_summary.description
资源 5 w（如在 DEFINE_MODE 中 5 w D）。 | G一 v 以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

mode_summary.sess_limit
1 0 会话限制。

mode_summary.act_sess_count
9 C 方 = D 活动会话总 }。如果已 + **active_sessions** 过K器 h 置为 AP_YES, 则C字段 + 一直大Z c。

mode_summary.fqplu_name
伙 i LU D 17 字Z全限定网 g { F。C { F G I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构I, " C EBCDIC U q R n d。(? v { F D $ 度最多I为 8 v字Z, " ; 含 6 入U q。）

mode_detail.overlay_size
C项中D字Z }, 即 = 下一 v 返回项（如果 P D话）D偏移 ?。

mode_detail.mode_name
为一组会话指定网 g t 性D方 = { 。 b G一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）, " 以 EBCDIC U q R n d。

mode_detail.description
资源 5 w（如在 DEFINE_MODE 中 5 w D）。

mode_detail.sess_limit
1 0 会话限制。

mode_detail.act_sess_count
9 C 方 = D 活动会话总 }。如果已 + **active_sessions** 过K器 h 置为 AP_YES, 则C字段 + 一直大Z c。

mode_detail.fqplu_name
伙 i LU D 17 字Z全限定网 g { F。C { F G I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构I, " C EBCDIC U q R n d。(? v { F D $ 度最多I为 8 v字Z, " ; 含 6 入U q。）

mode_detail.min_conwinners_source
指定 > X LU G y C $ 方（或“发话 _”）D会话D最小 } ?。

mode_detail.min_conwinners_target
指定 > X LU G y C ：方（或 “ k s _ ” ）D会话D最小 } ？。

mode_detail.drain_source
指定 1 | D或 4 位会话限制 1， > X LU G否在M放会话之 0 z 足H待会话 k s （AP_NO 或 AP_YES）。

mode_detail.drain_partner
指定 1 | D或 4 位会话限制 1， 伙 i LU G否在M放会话之 0 z 足H待会话 k s （AP_NO 或 AP_YES）。

mode_detail.auto_act
会话D | D } ？ k 伙 i LU ；换后自动激活D y C $ 方会话D } ？。

mode_detail.act_cw_count
9 C C方 = D活动D y C $ 方（或 “发话 _ ” ）会话D } ？。（ > X LU ；需要在 9 C某 v 会话之 0 x 行 k s 。）

mode_detail.act_cl_count
9 C C方 = D活动D y C ：方（或 “ k s _ ” ）会话D } ？。（ > X LU 在 9 C某 v 会话之 0 X须 x 行 k s 。）

mode_detail.sync_level
指定方 = y 支V D同 = 级（AP_NONE、AP_CONFIRM 或 AP_SYNCPT）。

mode_detail.default_ru_size
指定G否 9 C最大 RU 大小D缺！O限。如果C N } D值为 AP_YES, 则忽T在 **define_mode** O指定D **mode_chars.max_ru_size_upp** 字段, " R最大 RU 大小D O g ； h 置为 4 7 BTU 大小减去 TH 和 RH D大小。

AP_YES
AP_NO

mode_detail.max_neg_sess_limit
最大 I 协L D会话限制。指定方 = { D最大会话限制, > X LU I 以如目 j LU 一样在 CNOS 处理期间 9 C | 。

mode_detail.max_rcv_ru_size
最大 S U D RU 大小。

mode_detail.pending_session_count
指定暂挂（ H待会话激活D完 I ）会话D } ？。

mode_detail.termination_count
如果O一 v CNOS 动词 < 致方 = 会话限制 4 位为 0, 则 I 能 P 对话 9 C或H待 9 C b 些会话。 C字段D值为未；M放D会话D } ？。

mode_detail.implicit
指定项G否因为隐 =（AP_YES）定义或显 =（AP_NO）定义而放入。

如果因N } 错误而 9 动词；能执行, 则L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_MODE_NAME

AP_INVALID_PLU_NAME
AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_MODE_DEFINITION

QUERY_MODE_DEFINITION 返回以 0 在 DEFINE_MODE 动词O传入D信息和关Z SNA-定义D缺！方 = D信息。

b些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关某 v X定方 = D信息，或要获C几 v 『段』中D P m信息，& h置 **mode_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽T C字段。 k N阅Z12页D 『 i 询Z c 』，以获取关Z如何 9 C P m q = D 3 0 知 6 。

C P m y] **mode_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。

如果选中K AP_LIST_FROM_NEXT，则返回D P m 4 定义D次序（无 [指定D项G否存在）从下一项 * < 。

C动词 v 返回定义信息。C动词返回在 > X LU 和伙 i LU * < 9 C方 = 1 M确定D信息。

VCB a 构

```
typedef struct query_mode_definition
{
        unsigned short  opcode;             /* verb operation code          */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   format;             /* format                       */
        unsigned short  primary_rc;         /* primary return code          */
        unsigned long   secondary_rc;       /* secondary return code        */
        unsigned char   *buf_ptr;           /* pointer to buffer            */
        unsigned long   buf_size;           /* buffer size                  */
        unsigned long   total_buf_size;     /* total buffer size required   */
        unsigned short  num_entries;        /* number of entries            */
        unsigned short  total_num_entries;  /* total number of entries      */
        unsigned char   list_options;       /* listing options              */
        unsigned char   reserv3;            /* reserved                     */
        unsigned char   mode_name[8];       /* mode name                    */
} QUERY_MODE_DEFINITION;

typedef struct mode_def_summary
{
        unsigned short  overlay_size;       /* size of this entry           */
        unsigned char   mode_name[8];       /* mode name                    */
        unsigned char   description[RD_LEN];
                                            /* resource description         */
} MODE_DEF_SUMMARY;

typedef struct mode_def_detail
{
        unsigned short  overlay_size;       /* size of this entry           */
        unsigned char   mode_name[8];       /* mode name                    */
        MODE_CHARS      mode_chars;         /* mode characteristics         */
} MODE_DEF_DETAIL;

typedef struct mode_chars
{
        unsigned char   description[RD_LEN];
                                            /* resource description         */
        unsigned short  max_ru_size_upp;    /* max RU size upper bound      */
        unsigned char   receive_pacing_win; /* receive pacing window        */
        unsigned char   default_ru_size;    /* default RU size to maximize  */
                                            /* performance                  */
        unsigned short  max_neg_sess_lim;   /* max negotiable session limit */
```

```
        unsigned short  plu_mode_session_limit;
                                        /* LU-mode session limit        */
        unsigned short  min_conwin_src;     /* min source contention winner */
                                        /* sessions                     */
        unsigned char   cos_name[8];        /* class-of-service name        */
        unsigned char   cryptography;       /* cryptography                 */
        unsigned char   compression;        /* compression                  */
        unsigned short  auto_act;           /* initial auto-activation count*/
        unsigned short  min_conloser_src;   /* min source contention loser  */
        unsigned short  max_ru_size_low     /* maximum RU size lower bound  */
        unsigned short  max_receive_pacing_win;
                                        /* maximum receive pacing window*/
} MODE_CHARS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_MODE_DEFINITION

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **mode_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

mode_name

为一组会话指定网g t性D方={。 b G一v 8字Z字母}字D A型 EBCDIC字符串（以一v字母*头）， "以 EBCDIC U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**。

mode_def_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

mode_def_summary.mode_name
8 字Z D方={，|为一组会话指定网g X性。

mode_def_summary.description
资源5 w（如在 DEFINE_MODE 中5 w D）。|G一v以>X I显>字符集m >D 16 字Z字符串。y P 16 v字Z都P效。

mode_def_detail.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

mode_def_detail.mode_name
为一组会话指定网g t性D方={。b G一v 8 字Z字母 }字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC U q R n d。

mode_def_detail.mode_chars.description
资源5 w（如在 DEFINE_MODE 中5 w D）。|G一v以>X I显>字符集m >D 16 字Z字符串。y P 16 v字Z都P效。

mode_def_detail.mode_chars.max_ru_size_upp
要在P C方={ D会话O会话D最大 RU 大小D O限。

mode_def_detail.mode_chars.receive_pacing_win
指定1 9 C固定w = 1，会话D会话w=窗Z。指定1 9 C自J & w = 1， E先9 C D最小窗Z大小。

mode_def_detail.mode_chars.default_ru_size
指定G否9 C最大 RU 大小D缺！O限。如果C N }指定为 AP_YES，忽T **max_ru_size_upp**。

AP_YES
AP_NO

mode_def_detail.mode_chars.max_neg_sess_lim
最大I协L D会话限制。C Z协L在>X LU 和伙i LU 之间对Z指定方={ y允许D最大会话 }。

mode_def_detail.mode_chars.plu_mode_session_limit

在C方=下，最u协LD会话极限。C值m>W选会话极限，"CZ隐=CNOS。

范围: 0--32 767

mode_def_detail.mode_chars.min_conwin_src

II9C此方=D>X LU 激活DyC$方会话D最小}目。

范围: 0--32767

mode_def_detail.mode_chars.cos_name

服务级{F。bG一v 8字Z字母}字D A 型 EBCDIC 字符串（以一v字母*头），"以 EBCDIC UqRnd。

mode_def_detail.mode_chars.cryptography

指定G否在9CC方=D会话O9C加\（AP_NONE 或 AP_MANDATORY）。

mode_def_detail.mode_chars.compression

指定对激活会话（9C此方=）9C压u。

AP_COMP_PROHIBITED

在此方=会话O；支V RLE 压u。

AP_COMP_REQUESTED

在此方=会话O支V RLE 压u，"Rks RLE 压u（+；?制）。

mode_def_detail.mode_chars.auto_act

指定要自动激活Db种方=D会话}。C值CZ隐= CNOS。

范围: 0--32767

mode_def_detail.mode_chars.min_consloser_src

指定I任一此方=>X LU 激活DyC：方会话D最小}目。1 CNOS（会话；换}）；换隐=启动19C此值。

范围: 0--32767

mode_def_detail.mode_chars.max_ru_size_low

指定在此方=会话O发M和SU RU 最大大小D下g。1最大 RU 大小在会话激活期间协L1，9C此值。

范围: 0--61140

如果 **default_ru_size** h置为 AP_YES，则忽T此字段。

mode_def_detail.mode_chars.max_receive_pacing_win

指定此方=会话D会话最大w=窗Z。对Z自J&w=，此值CZ限制|ZhDSUw=窗Z。对Z固定w=，；9C此字段。注意，L序+<终9C自J&w=，}非相ZZc指定|；支V自J&w=。

范围: 0--32767

c值m>；POg。

如果因N}错误而9动词；能执行，则L序返回下PN}:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_MODE_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_MODE_TO_COS_MAPPING

QUERY_MODE_TO_COS_MAPPING 返回关Z方=至 COS D3dD信息。

b些信息以q=化Pm返回。如需P关某vX定方=D信息，或要获C几v『段』中DPm信息，&h置 **mode_name** 字段。

否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽TC字段。kN阅Z12页D『i询Zc』，以获取关Z如何9CPmq=D30知6。

CPmy] **mode_name** 排序。W先4{F$度排序，然后对相同$度D{Fx行ASCII 词d`辑排序（y] IBM D 6611 APPN MIB 定序）。如果选中KAP_LIST_FROM_NEXT，则返回DPm4定义D次序（无[指定D项G否存在）从下一项*<。

如果9C DEFINE_MODE 2G缺！ COS（未知方=3d=C COS），则QUERY_MODE_TO_COS_MAPPING 还返回 **mode_name** 为U（全c）D项和缺！COS。若x行排序，C项GZ一v。

VCB a构

```
typedef struct query_mode_to_cos_mapping
{
        unsigned short  opcode;            /* verb operation code        */
        unsigned char   reserv2;           /* reserved                   */
        unsigned char   format;            /* format                     */
        unsigned short  primary_rc;        /* primary return code        */
        unsigned long   secondary_rc;      /* secondary return code      */
        unsigned char   *buf_ptr;          /* pointer to buffer          */
        unsigned long   buf_size;          /* buffer size                */
        unsigned long   total_buf_size;    /* total buffer size required */
        unsigned short  num_entries;       /* number of entries          */
        unsigned short  total_num_entries; /* total number of entries    */
        unsigned char   list_options;      /* listing options            */
        unsigned char   reserv3;           /* reserved                   */
        unsigned char   mode_name[8];      /* mode name                  */
} QUERY_MODE_TO_COS_MAPPING;

typedef struct mode_to_cos_mapping_data
{
        unsigned short  overlay_size;      /* size of this entry         */
        unsigned char   mode_name[8];      /* mode name                  */
        unsigned char   cos_name[8];       /* COS name                   */
        unsigned char   reserva[20];       /* reserved                   */
} MODE_TO_COS_MAPPING_DATA;
```

提供N数

&CL序a供下PN}:

opcode

AP_QUERY_MODE_TO_COS_MAPPING

format

j6 VCB Dq=。+C字段h置为c，以指定OfPvD VCB Df>。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m > 无限制。

list_options

b m > 在P m信息中&返回2 4: 指定D **mode_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

mode_name

为一组会话指定网g t性D方={。 b G一v 8字Z字母}字D A型 EBCDIC字符串（以一v字母*头）， " 以 EBCDIC U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。 I h置为全c一m > 缺！ COS D项。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e x中返回D信息$度。

total_buf_size

返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。 C值I以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总}。 C值I以大Z **num_entries**。

mode_to_cos_mapping_data.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

mode_to_cos_mapping_data.mode_name

8字Z D方={， |为一组会话指定网g X性。如果C字段h置为全c，则m > 缺！ COS D项。

mode_to_cos_mapping_data.cos_name
k 方 = { 关 * D服务级 { F 。 b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_MODE_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_NMVT_APPLICATION

QUERY_NMVT_APPLICATION 返回一 v & C L 序D P m， b些& C L 序已 - 注 a K网 g 管 理 向 ? 传 M （ N M V T ） 级 消 息 （ 通 过 在 此 之 0 发 v REGISTER_NMVT_APPLICATION 动词）。 k N阅 Z 603页 D 『 Z 15 B 管理服务动词』以获取 | 详细 D 细 Z 。

C 信息作为 P m返回。如需 P 关某 v X 定& C L 序 D 信息，或要获 C 几 v 『段』中 D P m信息， & h 置 **application** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C 字段。 k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

VCB a 构

```
typedef struct query_nmvt_application
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* format                        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned char   *buf_ptr;           /* pointer to buffer             */
        unsigned long   buf_size;           /* buffer size                   */
        unsigned long   total_buf_size;     /* total buffer size required    */
        unsigned short  num_entries;        /* number of entries             */
        unsigned short  total_num_entries;  /* total number of entries       */
        unsigned char   list_options;       /* listing options               */
        unsigned char   reserv3;            /* reserved                      */
        unsigned char   application[8];     /* application                   */
} QUERY_NMVT_APPLICATION;

typedef struct nmvt_application_data
{
        unsigned short  overlay_size;       /* size of this entry            */
        unsigned char   application[8];     /* application name              */
        unsigned short  ms_vector_key_type; /* MS vector key accepted        */
                                            /* by appl                       */
        unsigned char   conversion_required;
                                            /* conversion to MDS_MU required */
        unsigned char   reserv[5];          /* reserved                      */
        unsigned char   reserva[20];        /* reserved                      */
} NMVT_APPLICATION_DATA;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_NMVT_APPLICATION

format

j 6 VCB D q = 。 + C 字段 h 置为 c ，以指定 O f P v D VCB D f > 。

buf_ptr

指向 C Z 写入 P m信息 D 缓 e x D 指 k 。 & C L 序 I + }] m加 = VCB D末尾（ X 须 + **buf_ptr** h 置为 NULL ）。

QUERY_NMVT_APPLICATION

buf_size

ｙａ供Ｄ缓ｅｘＤ大小。返回Ｄ｝］＋；，过ｂｖ大小。

num_entries

要返回Ｄ项Ｄ最大｝？。项Ｄ｝？；要，过ｂｖ值。ｃ值ｍ＞无限制。

list_options

ｂｍｗＰｍ信息中要返回２４：指定Ｄ **application**（见下ＰＮ｝）ｍ＞ｗ引值，Ｃｗ引值ＣＺ指定要返回Ｄ５际信息Ｄ起＜ｃ。

AP_FIRST_IN_LIST

忽Ｔｗ引值，返回Ｐｍ从ＰｍＤＺ一项＊＜。

AP_LIST_FROM_NEXT

返回Ｐｍ从Ｐｍ中４ａ供Ｄｗ引值ｙ指定Ｄ那项Ｄ下一项＊＜。

AP_LIST_INCLUSIVE

返回Ｐｍ从ｗ引值ｙ指定Ｄ那项＊＜。

application

＆ＣＬ序｛。｛ＦＧｖ ８字Ｚ字母｝字Ｄ Ａ型 EBCDIC 字符串或＿全EBCDIC ｃ。如果 **list_options** ｈ置为 AP_FIRST_IN_LIST，则忽ＴＣ字段。

返回Ｎ数

如果动词Ｉ功执行，则Ｌ序返回下ＰＮ｝：

primary_rc

AP_OK

buf_size

在缓ｅｘ中返回Ｄ信息＄度。

total_buf_size

返回值ｍｗ返回ｙＰ；ｋｓＤＰｍ信息ｙ需要Ｄ缓ｅｘＤ大小。Ｃ值Ｉ以大Ｚ **buf_size**。

num_entries

５际返回Ｄ项Ｄ｝？。

total_num_entries

已－返回Ｄ项Ｄ总｝。Ｃ值Ｉ以大Ｚ **num_entries**。

nmvt_application_data.overlay_size

Ｃ项中Ｄ字Ｚ｝，即＝下一ｖ返回项（如果ＰＤ话）Ｄ偏移？。

nmvt_application_data.application

注ａ＆ＣＬ序Ｄ｛Ｆ。｛ＦＧｖ ８字Ｚ字母｝字Ｄ Ａ型 EBCDIC 字符串。

nmvt_application_data.ms_vector_key_type

Ｉ＆ＣＬ序Ｓ＼Ｄ管理服务向？＼钥。１＆ＣＬ序注ａ NMVT 消息１，｜指定＆ＣＬ序＋Ｓ＼哪种管理服务向？＼钥。如需关Ｚ NMVT ＆ＣＬ序注ａＤ详细信息，ｋＮ阅Ｚ603页Ｄ『Ｚ15Ｂ 管理服务动词』。

nmvt_application_data.conversion_required

指定注ａ＆ＣＬ序Ｇ否要ｓ＋消息从 NMVT ｑ＝转换Ｉ MDS_MU ｑ＝

（AP_YES 或 AP_NO）。 1 & C L 序注 a NMVT 消息 1， | + 指定 G 否要 s 转换。如需关 Z NMVT & C L 序注 a D 详细信息， k N 阅 Z 603 页 D 『 Z 15 B 管理服务动词』。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_APPLICATION_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_NN_TOPOLOGY_NODE

C 动词 v J C Z 通信服务器。

？v 网 g Z c 维护网 g 拓扑 }] b， C }] b | 含网 g 中 D 网 g Z c、VRN 和网 g Z c 至网 g Z c TG D 信息。

QUERY_NN_TOPOLOGY_NODE 返回 C }] b 中 D 网 g Z c 和 VRN 项 D 信息。

b 些信息以 * 要 q = 或详细信息 q = D P m 返回。如需 P 关某 v X 定 Z c D 信息，或几 v 『段』中 D P m 信息，则 & C h 置 **node_name**、**node_type** 和 **frsn** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T b 些字段。k N 阅 Z 12 页 D 『i 询 Z c』，以获取关 Z 如何 9 C P m q = D 3 0 知 6。

C P m I **node_name**、**node_type** 和 **frsn** 组 I。先 4 { F $ 度排序 **node_name**，然后对相同 $ 度 D { F 4 ASCII 词 d ` 辑排序（y] IBM D 6611 APPN MIB 次序）。**node_type** 字段 4 下 P 次序排序: AP_NETWORK_NODE、AP_VRN。 **frsn** 4 } 字大小排序。

如果已选 AP_LIST_INCLUSIVE，则返回 D P m 从 C { F D Z 一 v P 效记 < * <。

如果选中 K AP_LIST_FROM_NEXT， P m + 从 { 为下 P 指定 { F 之一 D Z 一 v P 效记 < * <。

如果 **frsn** 字段（w 减 Y 序 P 号）h 置为非 c 值，则 v 返回 FRSN H | _ D }] b 项。| 允许先通过取 C Z c D 1 0 FRSN，而后以多“段”形 = 返回一致拓扑 }] b。| + 4 U 下 P Y 作 x 行:

1. 发 v QUERY_NODE，返回 Z c D 1 0 FRSN。
2. y] 需要发 v QUERY_NN_TOPOLOGY_NODE（FRSN h 置为 c），以取 C “段”中 D y P }] b 项。
3. 再次发 v QUERY_NODE， + 新 D FRSN k = 骤 1 中返回 D FRSN x 行 H O。
4. 如果 b = v FRSN ；同，则 m w }] b 已 P | D， y 以发 v QUERY_NN_TOPOLOGY_NODE， + FRSN h 置为 1，H 在 = 骤 1 中 a 供 D FRSN 大。

VCB a 构

```
typedef struct query_nn_topology_node
{
        unsigned short  opcode;            /* verb operation code          */
        unsigned char   reserv2;           /* reserved                     */
        unsigned char   format;            /* format                       */
        unsigned short  primary_rc;        /* primary return code          */
        unsigned long   secondary_rc;      /* secondary return code        */
        unsigned char   *buf_ptr;          /* pointer to buffer            */
        unsigned long   buf_size;          /* buffer size                  */
        unsigned long   total_buf_size;    /* total buffer size required   */
        unsigned short  num_entries;       /* number of entries            */
        unsigned short  total_num_entries; /* total number of entries      */
        unsigned char   list_options;      /* listing options              */
        unsigned char   reserv3;           /* reserved                     */
```

```
        unsigned char   node_name[17];     /* network qualified node name  */
        unsigned char   node_type;         /* node type                    */
        unsigned long   frsn;              /* flow reduction sequence num  */
} QUERY_NN_TOPOLOGY_NODE;
```

"：如果 **frsn** 字段 h 置为非 c 值，则 v 返回 FRSN 大 Z 指定值 D Z c 项。如果 h 置为 c，则返回 y P Z c 项。

```
typedef struct nn_topology_node_summary
{
        unsigned short  overlay_size;      /* size of this entry           */
        unsigned char   node_name[17];     /* network qualified node name  */
        unsigned char   node_type;         /* node type                    */
} NN_TOPOLOGY_NODE_SUMMARY;

typedef struct nn_topology_node_detail
{
        unsigned short  overlay_size;      /* size of this entry           */
        unsigned char   node_name[17];     /* network qualified node name  */
        unsigned char   node_type;         /* node type                    */
        unsigned short  days_left;         /* days left until entry purged */
        unsigned char   reserv1[2];        /* reserved                     */
        unsigned long   frsn;              /* flow reduction sequence num  */
        unsigned long   rsn;               /* resource sequence number     */
        unsigned char   rar;               /* route additional resistance  */
        unsigned char   status;            /* node status                  */
        unsigned char   function_support;  /* function support             */
        unsigned char   reserv2;           /* reserved                     */
        unsigned char   branch_aware;      /* node is branch aware         */
        unsigned char   reserva[20];       /* reserved                     */
} NN_TOPOLOGY_NODE_DETAIL;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_NN_TOPOLOGY_NODE

format

j 6 VCB D q =。+ C 字段 h 置为 c，以指定 O f P v D VCB D f >。

buf_ptr

指向 C Z 写入 P m 信息 D 缓 e x D 指 k。& C L 序 I + }] m 加 = VCB D 末尾（X 须 + **buf_ptr** h 置为 NULL）。

buf_size

y a 供 D 缓 e x D 大小。返回 D }] + ; ，过 b v 大小。

num_entries

要返回 D 项 D 最大 } ?。项 D } ? ; 要，过 b v 值。 c 值 m > 无限制。

list_options

m w 在 P m 信息中 & C 返回 2 4:

AP_SUMMARY

v 返回 * 要信息。

AP_DETAIL

返回详细信息。

指定 D **node_name**、**node_type** 和 **frsn** D 组合（见下 P N }）m > w 引值，C w 引值 C Z 指定要返回 D 5 际信息 D 起 < c。

AP_FIRST_IN_LIST
忽Tw引值，返回Pm从PmDZ一项*<。

AP_LIST_FROM_NEXT
返回Pm从Pm中4 a供Dw引值y指定D那项D下一项*<。

AP_LIST_INCLUSIVE
返回Pm从w引值y指定D那项*<。

node_name
来自网g拓扑}]bD网g全限定Zc{F。C{FD$度为 17 v字Z， IEBCDIC c串*D=v A型 EBCDIC 字符串组I， RI EBCDIC UqRnd。（?v{FD$度最多I为 8 v字Z， "；含6入Uq。）

node_type
ZcD类型。|Ih置为下P值之一:

AP_NETWORK_NODE
AP_VRN

如果；知@ **node_type**，则X须指定 AP_LEARN_NODE。

frsn w减Y序P号。如果|为非c值，则v返回 FRSN 大Z或HZC值DZc。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值mw返回yP；ksDPm信息y需要D缓e xD大小。C值I以大Z **buf_size**参数返回且EH。

42Tj /F44 1 Tf 3.6 -1Q0 TD (E

nn_topology_node_detail.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移？。

nn_topology_node_detail.node_name
来自网g拓扑}] b D网g全限定Z c { F。C { F D $度为 17 v字Z， I EBCDIC c串* D = v A 型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）

nn_topology_node_detail.node_type
Z c D类型。 | h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

nn_topology_node_detail.days_left
从拓扑}] b > } C Z c项之0 D l }。因为C项从未；> }， y以 | + 为> X Z c项h置为c。

nn_topology_node_detail.frsn
w减Y序P号。 | m w C资源在> X Z c O | 新D最后1间。

nn_topology_node_detail.rsn
资源序P号。 b I 5 P C资源D网g Z c分配。

nn_topology_node_detail.rar
Z c D 7 I =加阻9。

nn_topology_node_detail.status
指定Z c D状，。 C N } I以G AP_UNCONGESTED 或_ G下P一v或多v和 OR（或Y作）一起D值:

AP_CONGESTED
ISR 会话D }目大Z **isr_sessions_upper_threshold**。

AP_ERR_DEPLETED
端c会话D }目已达= y指定D最大值。

AP_IRR_DEPLETED
ISR 会话D }目已达=最大值。

AP_QUIESCING
已发v STOP_NODE 或类型 AP_QUIESCE 或 AP_QUIESCE_ISR

nn_topology_node_detail.function_support
指定支V哪些功能。 | I以G下P值中D一v或多v:

AP_PERIPHERAL BORDER_NODE
支V外围_ g Z c功能。

AP_EXTENDED BORDER_NODE
支V扩9 D _ g Z c功能。

AP_CDS
Z c支V主要D目<服务器功能。

AP_GATEWAY
Z c G网关Z c。（还未在a构O定义C功能。）

AP_INTERCHANGE_NODE
C Z c G网关Z c。(还未在a构O定义C功能。)

AP_ISR
Z c支V中间会话7 I选择。

AP_HPR
Z c支V _性能7 I选择D基>功能。

AP_RTP_TOWER
Z c支V HPR D RTP ~。

AP_CONTROL_OVER_RTP_TOWER
Z c支V RTP ~ O D X制w。

": AP_CONTROL_OVER_RTP_TOWER k AP_HPR 和 AP_RTP_TOWER D h置相符。

nn_topology_node_detail.branch_aware
5 w Z c G否G分支。

AP_NO
C Z c ;G分支。

AP_YES
C Z c G分支。

如果因N }错误而9动词;能执行, 则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_NODE

AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词;能执行, 则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词;能执行, 则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_NN_TOPOLOGY_STATS

C动词v J C Z通信服务器。

QUERY_NN_TOPOLOGY_STATS 返回P关拓扑}] b D统计信息， " R v在网g Z c处发v。

VCB a构

```
typedef struct query_nn_topology_stats
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* format                        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned long   max_nodes;          /* max num of nodes in database  */
        unsigned long   cur_num_nodes;      /* current number of nodes in    */
                                            /* database                      */
        unsigned long   node_in_tdus;       /* number of TDUs received       */
        unsigned long   node_out_tdus;      /* number of TDUs sent           */
        unsigned long   node_low_rsns;      /* node updates received with    */
                                            /* low RSNs                      */
        unsigned long   node_equal_rsns;    /* node updates in with          */
                                            /* equal RSNs                    */
        unsigned long   node_good_high_rsns;
                                            /* node updates in with          */
                                            /* high RSNs                     */
        unsigned long   node_bad_high_rsns;
                                            /* node updates in with          */
                                            /* high and odd RSNs             */
        unsigned long   node_state_updates; /* number of node updates sent   */
        unsigned long   node_errors;        /* number of node entry          */
                                            /* errors found                  */
        unsigned long   node_timer_updates; /* number of node records built  */
                                            /* due to timer updates          */
        unsigned long   node_purges;        /* num node records purged       */
        unsigned long   tg_low_rsns;        /* TG updates received with      */
                                            /* low RSNs                      */
        unsigned long   tg_equal_rsns;      /* TG updates in with equal RSNs */
        unsigned long   tg_good_high_rsns;  /* TG updates in with high RSNs  */
        unsigned long   tg_bad_high_rsns;   /* TG updates in with high       */
                                            /* and odd RSNs                  */
        unsigned long   tg_state_updates;   /* number of TG updates sent     */
        unsigned long   tg_errors;          /* number of TG entry errors     */
                                            /* found                         */
        unsigned long   tg_timer_updates;   /* number of node records        */
                                            /* built due to timer updates    */
        unsigned long   tg_purges;          /* num node records purged       */
        unsigned long   total_route_calcs;  /* num routes calculated for COS */
        unsigned long   total_route_rejs;   /* num failed route calculations */
        unsigned long   total_tree_cache_hits;
                                            /* total num of tree cache hits  */
        unsigned long   total_tree_cache_misses;
                                            /* total num of tree cache       */
                                            /* misses                        */
        unsigned counter
                        total_tdu_wars;    /* total number TDU war           */
        unsigned char   reserva[16];       /* reserved                     */
} QUERY_NN_TOPOLOGY_STATS;
```

QUERY_NN_TOPOLOGY_STATS

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_NN_TOPOLOGY_STATS

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

max_nodes

拓扑}] b中Z c记< D最大} ?（c m >无限制)。

cur_num_nodes

C Z c D拓扑}] b中D Z c D 1 0号。如果C值，过允许D Z c D最大号，则发v / (。

node_in_tdus

C Z c y S U D拓扑}] b |新(TDU)D总}。

node_out_tdus

自O一次u <化以来，I C Z c (" R要发M = y P Z |网g Z c D D拓扑}] b |新(TDU)D总}。

node_low_rsns

C Z c C H 1 0 RSN M D RSN S U D拓扑Z c |新D总}。偶}和奇}D RSN 都|括在C计}值中。(b些拓扑}] b |新(TDU)；G错误，+在TDU向y P Z |网g Z c广%1 z z。C Z c D拓扑}] b未发z |新，+C Z c C其O _ D RSN + TDU 发M = 发M C M RSN D Z | Z c。)

node_equal_rsns

C Z c C k 1 0 RSN 相H D RSN S U D拓扑Z c |新D总}。偶}和奇}D RSN 都|括在C计}值中。(b些拓扑}] b |新(TDU)；G错误，+在TDU向y P Z |网g Z c广%1 z z。C Z c拓扑}] b未发z |新。)

node_good_high_rsns

C Z c C H 1 0 RSN _ D RSN S U D拓扑Z c |新D总}。C Z c |新其拓扑R Q TDU 向y P Z |网g Z c广%。；需要向C |新D发M方发M TDU，因为Z c已 - |新K。

node_bad_high_rsns

C Z c C H 1 0 RSN _ D奇} RSN S U D拓扑Z c |新D总}。b些|新m >一v APPN 网g Z c检b = K拓扑D；一致。C Z c |新其拓扑R Q TDU向y P Z |网g Z c广%。

node_state_updates

拓扑Z c |新D总}，C |新作为0响 APPN 拓扑和7 I选择D内?检b Z c状，|D D a果。TDU Q |新发M至y P Z |网g Z c。

node_errors

C Z c检b = D拓扑Z c | 新；一致D总 } 。 b在C Z c T图 | 新其拓扑 }] b " 检b = }] ；一致1发z 。 C Z c C] 增为下一v奇 } D 1 0 RSN 创 (TDU, " Q | 向y P Z | 网g Z c广%。

node_timer_updates

I Z计1器 | 新而为C Z c资源 (" D拓扑Z c | 新D总 } 。 TDU Q | 新发M至y P Z | 网g Z c 。 b些 | 新#证其 | 网g Z c ；从 | G D拓扑 }] b中> } C Z c D资源。

node_purges

从C Z c D拓扑 }] b中e } D拓扑Z c记< D总 } 。在指定1间内未 | 新Z c记< 1发z b种i v 。支配Z c对其要# V在网g拓扑中D资源D广% | 新：责。

tg_low_rsns

C Z c C H 1 0 RSN M D RSN S U D拓扑 TG | 新D总 } 。偶 } 和奇 } D RSN 都 | 括在C计 } 值中。 （ b些拓扑 }] b | 新(TDU) ; G错误, +在 TDU向y P Z | 网g Z c广% 1 z z 。 C Z c D拓扑 }] b未发z | 新, + C Z c C其O _ D RSN + TDU 发M = 发M C M RSN D Z | Z c 。）

tg_equal_rsns

C Z c C k 1 0 RSN 相H D RSN S U D拓扑 TG | 新D总 } 。偶 } 和奇 } D RSN 都 | 括在C计 } 值中。 （ b些拓扑 }] b | 新(TDU) ; G错误, +在 TDU 向y P Z | 网g Z c广% 1 z z 。 C Z c拓扑 }] b未发z | 新。）

tg_good_high_rsns

C Z c C H 1 0 RSN _ D RSN S U D拓扑 TG | 新D总 } 。 C Z c | 新其拓扑R Q TDU 向y P Z | 网g Z c广%。

tg_bad_high_rsns

C Z c C H 1 0 RSN _ D奇 } RSN S U D拓扑 TG | 新D总 } 。 b些 | 新m > 一v APPN 网g Z c检b = K拓扑D ；一致。 C Z c | 新其拓扑R Q TDU向y P Z | 网g Z c广%。

tg_state_updates

拓扑 TG | 新D总 }, | 作为0响 APPN 拓扑和7 I 选择D内？检b Z c状，| D D a果。TDU Q | 新发M至y P Z | 网g Z c 。

tg_errors

C Z c检b = D拓扑 TG | 新；一致D总 } 。 b在C Z c T图 | 新其拓扑 }] b " 检b = }] ；一致1发z 。 C Z c C] 增为下一v奇 } D 1 0 RSN 创 (TDU, " Q | 向y P Z | 网g Z c广%。

tg_timer_updates

I Z计1器 | 新而为C Z c资源 (" D拓扑 TG | 新D总 } 。TDU Q | 新发M至y P Z | 网g Z c 。 b些 | 新#证其 | 网g Z c ；从 | G D拓扑 }] b中> } C Z c D资源。

tg_purges

从C Z c D拓扑 }] b中e } D拓扑 TG 记< D总 } 。在指定1间内未 | 新Z c记< 1发z b种i v 。支配Z c对其要# V在网g拓扑中D资源D广% | 新：责。

QUERY_NN_TOPOLOGY_STATS

total_route_calcs

从最后一次 * <，为 y P 服务级计 c D 7 6 } 。

total_route_rejs

自从 O 一次 u < 化以来；能计 c D y P 服务级 D 7 6 k s } 。

total_tree_cache_hits

_ Y 缓存 7 I 选择 w z 足 D 7 6 计 c } 。注意， I Z ? v 7 6 I 能需要检 i 几 v w， y 以 C } I 能 H 计 c D 7 6 总 } 大。

total_tree_cache_misses

_ Y 缓存 7 I 选择 w ； z 足 D 7 6 计 c } ， y 以 X 须 (" 新 D 7 I 选择 w 。

total_tdu_wars

> X Z c 已检 b = D 和已预防 D TDU e 突 D } ? 。

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_NN_TOPOLOGY_TG

C动词ｖＪＣＺ通信服务器。

？ｖ网ｇＺｃ维护网ｇ拓扑｝］ｂ，Ｃ｝］ｂ｜含网ｇ中D网ｇＺｃ、VRN 和网ｇＺｃ至网ｇＺｃ TG D信息。 QUERY_NN_TOPOLOGY_TG 返回Ｐ关Ｃ｝］ｂ中D TG 项D信息。

ｂ些信息以＊要ｑ＝或详细信息ｑ＝ＤＰｍ返回。如需关Ｚ某ｖＸ定ＺｃＤ信息，或获取几ｖ『ｉ』中ＤＰｍ信息，则＆Ｃｈ置 **owner**、**owner_type**、 **dest**、**dest_type**、**tg_num** 和 **frsn**。否则（如果 **list_options** 字段ｈ置为 AP_FIRST_IN_LIST），忽Ｔｂ些字段。ｋＮ阅Ｚ12页Ｄ『ｉ询Ｚｃ』，以获取关Ｚ如何９ＣＰｍｑ＝Ｄ３０知６。

ＣＰｍＩ **owner**、 **owner_type**、**dest**、**dest_type**、**tg_num** 和 **frsn** 组Ｉ。先４｛Ｆ＄度排序 **owner** ｛Ｆ和 **dest** ｛Ｆ，然后对相同＄度Ｄ｛Ｆ４ ASCII 词ｄ｀辑排序（ｙ］ IBM Ｄ 6611 APPN MIB 次序）。**owner_type** 和 **dest_type** ４下Ｐ次序排序: AP_NETWORK_NODE、AP_VRN。**tg_num** 和 **frsn** ４｝字大小排序

寸｛ 选 APLI

```
        unsigned char   owner_type;        /* type of node that owns the TG*/
        unsigned char   dest[17];          /* TG destination node          */
        unsigned char   dest_type;         /* TG destination node type     */
        unsigned char   tg_num;            /* TG number                    */
        unsigned char   reserv1;           /* reserved                     */
        unsigned long   frsn;              /* flow reduction sequence num  */
} QUERY_NN_TOPOLOGY_TG;

typedef struct topology_tg_summary
{
        unsigned short  overlay_size;      /* size of this entry           */
        unsigned char   owner[17];         /* node that owns the TG        */
        unsigned char   owner_type;        /* type of node that owns the TG*/
        unsigned char   dest[17];          /* TG destination node          */
        unsigned char   dest_type;         /* TG destination node type     */
        unsigned char   tg_num;            /* TG number                    */
        unsigned char   reserv3[1];        /* reserved                     */
        unsigned long   frsn;              /* flow reduction sequence num  */
} TOPOLOGY_TG_SUMMARY;

typedef struct topology_tg_detail
{
        unsigned short  overlay_size;      /* size of this entry           */
        unsigned char   owner[17];         /* node that owns the TG        */
        unsigned char   owner_type;        /* type of node that owns the TG*/
        unsigned char   dest[17];          /* TG destination node          */
        unsigned char   dest_type;         /* TG destination node type     */
        unsigned char   tg_num;            /* TG number                    */
        unsigned char   reserv3[1];        /* reserved                     */
        unsigned long   frsn;              /* flow reduction sequence num  */
        unsigned short  days_left;         /* days left until entry purged */
        LINK_ADDRESS    dlc_data           /* DLC signalling data          */
        unsigned long   rsn;               /* resource sequence number     */
        unsigned char   status;            /* node status                  */
        TG_DEFINED_CHARS tg_chars;         /* TG characteristics           */
        unsigned char   subarea_number[4];
                                           /* subarea number               */
        unsigned char   tg_type;           /* TG type                      */
        unsigned char   intersubnet_tg;    /* intersubnet TG?              */
        unsigned char   cp_cp_session_active;
                                           /* CP-CP session is active      */
        unsigned char   branch_tg;         /* TG is a branch TG            */
        unsigned char   reserva[12];       /* reserved                     */
} TOPOLOGY_TG_DETAIL;

typedef struct link_address
{
        unsigned short  length;             /* length                      */
        unsigned short reserve1;            /* reserved                    */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                            /* address                     */
} LINK_ADDRESS;
```

"： 如果 **frsn** 字段 h 置为非 c 值，则 v 返回 C FRSN D Z c 项。如果 h 置为 c，则返回 y P Z c 项。

提供N数

& C L 序 a 供下 P N }：

opcode

AP_QUERY_NN_TOPOLOGY_TG

format

j 6 VCB D q =。+ C 字段 h 置为 c，以指定 O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓 e x D指 k。 &C L序 I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓 e x D大小。返回D }] + ; ，过 b v大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过 b v值。 c 值m > 无限制。

list_options

m w在P m信息中&C返回 2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **owner**、**owner_type**、**dest**、 **dest_type**、**tg_num** 和 **frsn** D组合（见下P N }）m > w引值， C w引值C Z 指定要返回D 5 际信息D起 < c 。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z 一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中 4 a 供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值 y 指定D那项 * <。

owner TG D源Z c { F。C { F D $ 度为 17 v字Z， I EBCDIC c串 * D = v A 型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（? v { F D $ 度最多I为 8 v字Z， " ；含 6 入U q 。）如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

owner_type

5 P TG D Z c 类型。 | I h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

如果；知@ **owner_type**，则X须指定 AP_LEARN_NODE。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

dest TG D全限定目D Z c { 。C { F D $ 度为 17 v字Z， I EBCDIC c串 * D = v A 型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（? v { F D $ 度最多I为 8 v字Z， " ；含 6 入U q 。）如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

dest_type

C TG D目D Z c 类型。 | I h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

如果；知@ **dest_type**，则X须指定 AP_LEARN_NODE。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

tg_num
k TG 相关D号k。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

frsn w减Y序P号。如果|为非c值，则v返回 FRSN 大Z或H Z C值D Z c。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总}。C值I以大Z **num_entries**。

topology_tg_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

topology_tg_summary.owner
TG D源Z c { F。C { F D $度为 17 v字Z， I EBCDIC c串* D = v A型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（? v { F D $度最多I为 8 v字Z， " ；含6入U q。）

topology_tg_summary.owner_type
5 P TG D Z c类型。| h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_summary.dest
TG D全限定目D Z c {。C { F D $度为 17 v字Z， I EBCDIC c串* D = v A型 EBCDIC 字符串组I， R I EBCDIC U q R n d。（? v { F D $度最多I为 8 v字Z， " ；含6入U q。）

topology_tg_summary.dest_type
C TG D目D Z c类型。| h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_summary.tg_num
k TG 相关D号k。

topology_tg_summary.frsn
w减Y序P号。 | m w C资源在> X Z c O | 新D最后1间。

topology_tg_detail.overlay_size
C项中D字Z }, 即=下一v返回项（如果P D话）D偏移？。

topology_tg_detail.owner
TG D源Z c { F。C { F D $度为 17 v字Z, I EBCDIC c串* D = v A型 EBCDIC 字符串组I, R I EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z, " ；含6人U q。）

topology_tg_detail.owner_type
5 P TG D Z c类型。 | h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_detail.dest
TG D全限定目D Z c { 。C { F D $度为 17 v字Z, I EBCDIC c串* D = v A型 EBCDIC 字符串组I, R I EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z, " ；含6人U q。）

topology_tg_detail.dest_type
C TG D目D Z c类型。 | h置为下P值之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_detail.tg_num
k TG 相关D号k。

topology_tg_detail.frsn
w减Y序P号。 | m w C资源在> X Z c O | 新D最后1间。

topology_node_detail.days_left
从拓扑}] b > } C Z c项之0 D l }。

topology_tg_detail.dlc_data.length
= VRN D, S D DLC X址$度（如果 **dest_type** ；G AP_VRN, 则h置为c)。 .

topology_tg_detail.dlc_data.address
= VRN D, S D DLC X址。如果 **dest_type** ；为 AP_VRN, 则+ | h置为c。 .

topology_tg_detail.rsn
资源序P号。 b I 5 P C资源D网g Z c分配。

topology_tg_detail.status
指定 TG D状，。C N } I以G下P一v或多v和 OR（或Y作）一起D值:

AP_TG_OPERATIVE
AP_TG_QUIESCING
AP_TG_GARBAGE_COLLECT
AP_TG_CP_CP_SESSIONS

AP_TG_HPR
AP_TG_RTP
AP_TG_NONE

topology_tg_detail.tg_chars
TG X性。

topology_tg_detail.subarea_number
如果 TG D 5 P _或目D Z c _ P子x，则C字段 | 含 4 或 5 型Z c D子x }， b v Z c 5 P k _ P子x D Z c O D TG 相关D 4 7 >。否则， C字段+ ； h 置为全二x 制 c 。

topology_tg_detail.tg_type
TG 类型。 C字段取下P 其中一 v 值:

AP_APPN_OR_BOUNDARY_TG
APPN TG 或 boundary-function-based TG

AP_INTERCHANGE_TG
；换 TG

AP_VIRTUAL_ROUTE_BASED_TG
Virtual-route-based TG

AP_UNKNOWN
在拓扑中 (f D C TG D TG 类型未知。

topology_tg_detail.intersubnet.tg
C TG G ; G互，子网 TG ？

AP_YES
AP_NO

topology_tg_detail.cp_cp_session_active
指定 5 P Z c D : y $ 方 CP-CP 会话G否激活(AP_UNKNOWN、AP_NO 或 AP_YES)。

branch_tg
5 w TG G否G分支 TG。

AP_NO
TG ；G分支 TG。

AP_YES
TG G分支 TG。

如果因N } 错误而 9 动词；能执行，则L 序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_TG

AP_INVALID_ORIGIN_NODE
AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_NODE

QUERY_NODE 返回 Z c D X定信息和统计。 } K返回在执行期间动，确定D信息外, QUERY_NODE 还返回在 Z c u < 化期间 h 置 D N } 。

VCB a 构

格式 2

```
typedef struct query_node
{
        unsigned short  opcode;             /* verb operation code          */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   format;             /* format                       */
        unsigned short  primary_rc;         /* primary return code          */
        unsigned long   secondary_rc;       /* secondary return code        */
        CP_CREATE_PARMS cp_create_parms;    /* create parameters            */
        unsigned long   up_time;            /* time since node started      */
        unsigned long   mem_size;           /* size of memory available     */
        unsigned long   mem_used;           /* size of memory used          */
        unsigned long   mem_warning_threshold;
                                            /* memory constrained           */
                                            /* threshold                    */
        unsigned long   mem_critical_threshold;
                                            /* memory critical threshold    */
        unsigned char   nn_functions_supported;
                                            /* NN functions supported       */
        unsigned char   functions_supported;
                                            /* functions supported          */
        unsigned char   en_functions_supported;
                                            /* EN functions supported       */
        unsigned char   nn_status;          /* node status.  One or more of */
        unsigned long   nn_frsn;            /* NN flow reduction            */
                                            /* sequence number              */
        unsigned long   nn_rsn;             /* Resource sequence number     */
        unsigned short  def_ls_good_xids;   /* Good XIDs for defined        */
                                            /* link stations                */
        unsigned short  def_ls_bad_xids;    /* Bad XIDs for defined         */
                                            /* link stations                */
        unsigned short  dyn_ls_good_xids;   /* Good XIDs for dynamic        */
                                            /* link stations                */
        unsigned short  dyn_ls_bad_xids;    /* Bad XIDs for dynamic         */
                                            /* link stations                */
        unsigned char   dlur_release_level; /* Current DLUR release level   */
        unsigned char   nns_dlus_served_lu_reg_supp;
                                            /* NNS support for registration */
                                            /* of DLUS-served LUs reserved  */
      unsigned char   reserva[19];          /* reserved                     */
      unsigned char   fq_nn_server_name[17];                                  */
                                            /* FQ name of NN server         */
      unsigned long   current_isr_sessions;/* current ISR sessions         */
      unsigned char   nn_functions2;        /* NN functions continued       */
      unsigned char   branch_ntwk_arch_version;
                                            /* branch network architecture  */
                                            /* version supported            */
      unsigned char   reservb[28];          /* reserved                     */
} QUERY_NODE;

typedef struct cp_create_parms
{

/*reserved                    */
```

```
        unsigned char   cp_alias[8];        /* CP alias                    */


        unsigned char   mode_to_cos_map_supp;
                                            /* mode to COS mapping support  */
        unsigned char   mds_supported;      /* MDS and MS capabilities      */
        unsigned char   node_id[4];         /* node ID                      */
        unsigned short  max_locates;        /* max locates node can process */
        unsigned short  dir_cache_size;     /* directory cache size         */
                                            /* (reserved) if not NN)        */
        unsigned short  max_dir_entries;    /* max directory entries        */
        unsigned short  locate_timeout;     /* locate timeout in seconds    */
        unsigned char   reg_with_nn;        /* register resources with NNS  */
        unsigned char   reg_with_cds;       /* resource registration with   */
                                            /* CDS                          */
        unsigned short  mds_send_alert_q_size;
                                            /* size of MDS send alert queue */
        unsigned short  cos_cache_size;     /* number of COS definitions    */
        unsigned short  tree_cache_size;    /* Topology Database routing    */
                                            /* tree cache size              */
        unsigned short  tree_cache_use_limit;
                                            /* num times tree can be used   */
        unsigned short  max_tdm_nodes;      /* max num nodes that can be    */
                                            /* stored in Topology Database  */
        unsigned short  max_tdm_tgs;        /* max num TGs that can be      */
                                            /* stored in Topology Database  */
        unsigned long   max_isr_sessions;   /* max ISR sessions             */
        unsigned long   isr_sessions_upper_threshold;
                                            /* upper threshold for ISR sess */
        unsigned long   isr_sessions_lower_threshold;
                                            /* lower threshold for ISR sess */
        unsigned short  isr_max_ru_size;    /* max RU size for ISR          */
        unsigned short  isr_rcv_pac_window; /* ISR rcv pacing window size   */
        unsigned char   store_endpt_rscvs;  /* endpoint RSCV storage        */
        unsigned char   store_isr_rscvs;    /* ISR RSCV storage             */
        unsigned char   store_dlur_rscvs;   /* DLUR RSCV storage            */
        unsigned char   dlur_support;       /* is DLUR supported?           */
        unsigned char   pu_conc_support;    /* is PU conc supported?        */
        unsigned char   nn_rar;             /* Route additional resistance  */
        unsigned char   hpr_support;        /* level of HPR support         */
        unsigned char   mobile;             /* HPR path-switch controller?  */
        unsigned char   discovery_support;  /* Discovery function utilized  */
        unsigned char   discovery_group_name[8];
                                            /* Group name for Discovery     */
        unsigned char   implicit_lu_0_to_3;
                                            /* Implicit LU 0 to 3 support   */
        unsigned char   default_preference;
                                            /* Default routing preference   */
        unsigned char   anynet_supported;
                                            /* level of AnyNet support      */
        unsigned short  max_ls_exception_events;
                                            /* maximum LS Exception events  */
        unsigned char   comp_in_series;     /* compression in series allowed*/
        unsigned char   max_compress_lvl;   /* maximum compression level    */
        unsigned char   node_spec_data_len; /* length of node specific data */
        unsigned char   ptf[64];            /* program temporary fix array  */
} CP_CREATE_PARMS;
```

格式 1（后备 6 别）

```
typedef struct query_node
{
        unsigned short  opcode;             /* verb operation code          */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   format;             /* format                       */
        unsigned short  primary_rc;         /* primary return code          */
        unsigned long   secondary_rc;       /* secondary return code        */
```

```
        CP_CREATE_PARMS cp_create_parms;    /* create parameters            */
        unsigned long   up_time;            /* time since node started      */
        unsigned long   mem_size;           /* size of memory available     */
        unsigned long   mem_used;           /* size of memory used          */
        unsigned long   mem_warning_threshold;
                                            /* memory constrained           */
                                            /* threshold                    */
        unsigned long   mem_critical_threshold;
                                            /* memory critical threshold    */
        unsigned char   nn_functions_supported;
                                            /* NN functions supported       */
        unsigned char   functions_supported;
                                            /* functions supported          */
        unsigned char   en_functions_supported;
                                            /* EN functions supported       */
        unsigned char   nn_status;          /* node status.  One or more of */
        unsigned long   nn_frsn;            /* NN flow reduction            */
                                            /* sequence number              */
        unsigned long   nn_rsn;             /* Resource sequence number     */
        unsigned short  def_ls_good_xids;   /* Good XIDs for defined        */
                                            /* link stations                */
        unsigned short  def_ls_bad_xids;    /* Bad XIDs for defined         */
                                            /* link stations                */
        unsigned short  dyn_ls_good_xids;   /* Good XIDs for dynamic        */
                                            /* link stations                */
        unsigned short  dyn_ls_bad_xids;    /* Bad XIDs for dynamic         */
                                            /* link stations                */
        unsigned char   dlur_release_level; /* Current DLUR release level   */
        unsigned char   reserva[19];         /* reserved                     */
} QUERY_NODE;
```

格式 0（后备 6 别）

```
typedef struct query_node
{
        unsigned short  opcode;             /* verb operation code          */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   format;             /* format                       */
        unsigned short  primary_rc;         /* primary return code          */
        unsigned long   secondary_rc;       /* secondary return code        */
        CP_CREATE_PARMS cp_create_parms;    /* create parameters            */
        unsigned long   up_time;            /* time since node started      */
        unsigned long   mem_size;           /* size of memory available     */
        unsigned long   mem_used;           /* size of memory used          */
        unsigned long   mem_warning_threshold;
                                            /* memory constrained           */
                                            /* threshold                    */
        unsigned long   mem_critical_threshold;
                                            /* memory critical threshold    */
        unsigned char   nn_functions_supported;
                                            /* NN functions supported       */
        unsigned char   functions_supported;
                                            /* functions supported          */
        unsigned char   en_functions_supported;
                                            /* EN functions supported       */
        unsigned char   nn_status;          /* node status.  One or more of */
        unsigned long   nn_frsn;            /* NN flow reduction            */
                                            /* sequence number              */
        unsigned long   nn_rsn;             /* Resource sequence number     */
        unsigned short  def_ls_good_xids;   /* Good XIDs for defined        */
                                            /* link stations                */
        unsigned short  def_ls_bad_xids;    /* Bad XIDs for defined         */
                                            /* link stations                */
        unsigned short  dyn_ls_good_xids;   /* Good XIDs for dynamic        */
                                            /* link stations                */
        unsigned short  dyn_ls_bad_xids;    /* Bad XIDs for dynamic         */
```

```
                                        /* link stations                */
        unsigned char   dlur_release_level; /* Current DLUR release level   */
        unsigned char   reserva[19];        /* reserved                     */
} QUERY_NODE;
```

提供N数

＆C L序a供下P N }:

opcode

AP_QUERY_NODE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

1 C字段h置为c 1，下P D v字段为 Unsigned short: **def_ls_good_xids**、**def_ls_bad_xids**、**dyn_ls_good_xids** 和 **dyn_ls_bad_xids**，而；G Unsigned_COUNTER。

1 C字段h置为 2 1， 4 U y h v D那样来9 C下P字段: **fq_nn_server_name** 和 **current_isr_sessions**。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

cp_create_parms.crt_parms_len

创(N } a构D $度。

cp_create_parms.description

资源5 w。 | G一v以> X I显>字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

cp_create_parms.node_type

| 通# G:

AP_END_NODE
AP_NETWORK_NODE
AP_LEN_NODE
AP_BRANCH_NETWORK_NODE

cp_create_parms.fqcp_name

Z c D 17 字Z全限定X制c { F。C { F G I = v C EBCDIC c串* D A 型EBCDIC 字符串构I， " C EBCDIC U q R n d。(？v { F D $度最多I为8 v字Z， " ；含6人U q。)

cp_create_parms.cp_alias

> X 9 C D X制c p {。 | G在> X I显>字符集中D一v 8 字Z字符串。 y P D 8 v字Z G P效D。

cp_create_parms.mode_to_cos_map_supp

指定Z c G否支V方= 至 COS D 3 d (AP_YES 或 AP_NO)。如果+ | h置为

AP_YES，则在 DEFINE_MODE 动词O指定D COS X须G一v SNA 定义D COS，或已通过发v DEFINE_COS 动词定义。

cp_create_parms.mds_supported

指定管理服务G否支V“多v r 支V”和“管理服务能&”(AP_YES 或 AP_NO)。

cp_create_parms.node_id

在 XID ；换中9CDZc j 6符。|G一v 4 字Z . y x制字符串。

cp_create_parms.max_locates

Z c I处理D定位D最大}。

cp_create_parms.dir_cache_size

v J C Z网g Z c: 目< _ Y缓存大小。

cp_create_parms.max_dir_entries

目<项D最大}目。如果C字段h置为c，则|G；\限制D。

cp_create_parms.locate_timeout

指定以k为%位D网g Q w，1之0D1间。 0 m>Q w无，1。

cp_create_parms.reg_with_nn

指定G否C网g Z c服务器注a资源。注a ' \ ；0响Z c u <化D I功完I。k N阅Z539页D『REGISTRATION_FAILURE』，以获取|多信息。EN 和 BrNN 对C字段P；同D b M。

末端Z c:

AP_NO

C Z c未9C|D NN 服务器注a任何 LU。NNS + y P广%Q w转发x端Z c。

AP_YES

C Z c C|D NNS 注a y P>X从t（如果 NNS 支V选项集 1116）和y P>X独" LU。 NNS v +直S位置转M x|(}非|5P；能注a D从t LU)。

分支网g Z c:

AP_REGISTER_NONE

C Z c未9C|D NN 服务器注a任何 LU。

AP_REGISTER_ALL

C Z c C|D NNS 注a y P>X从t（如果|支V DLUR 全多子网，R NNS 支V选项集 1116）和y P r独" LU。

AP_REGISTER_LOCAL_ONLY

C Z c C|D NNS 注a y P>X从t（如果|支V DLUR 全多子网，R NNS 支V选项集 1116）和y P>X独" LU。

cp_create_parms.reg_with_cds

指定G否允许C中央目<服务器(CDS)注a资源。 EN、NN 或 BrNN 对C字段P；同D b M。

端Z c: 指定G否允许C CDS 端Z c资源注a NNS。如果 **reg_with_nn** h置为 AP_NONE，则忽T C字段。

AP_NO

；能C CDS 注a EN 资源。

AP_YES

I以C CDS 注a EN 资源。

网g Z c: 指定G否I以C CDS 注a > X资源和r资源（5 P EN 允许C CDS 注a）。

AP_NO

；能C CDS 注a > X或r资源。

AP_YES

I以C CDS 注a > X或r资源。注a ' \ ；0响 START_NODE 动词D I功完I。

分支网g Z c: 指定G否允许C CDS BrNN 资源注a NNS （= BrNN 或来自 BrNN D r）。如果 **reg_with_nn** h置为 AP_NO, 则忽T C字段。

AP_REGISTER_NONE

C Z c未9 C | D NN 服务器注a任何 LU。

AP_REGISTER_ALL

C Z c C | D NNS 注a y P > X从t（如果|支V DLUR 全多子网, R NNS 支V选项集 1116）和y P r独" LU。

AP_REGISTER_LOCAL_ONLY

C Z c C | D NNS 注a y P > X从t（如果|支V DLUR 全多子网, R NNS 支V选项集 1116）和y P > X独" LU。

cp_create_parms.mds_send_alert_q_size

MDS 发M / (队P D大小。1达= C限制1, MDS 组件+ > }队P中最老D项。

cp_create_parms.cos_cache_size

COS }] b加权_ Y缓存大小。

cp_create_parms.tree_cache_size

拓扑}] b 7 I选择w _ Y缓存D大小。

cp_create_parms.tree_cache_use_limit

9 C _ Y缓存w D最大}。一)，过C }, + Q w废弃, "重新计c。b允许Z c在相H D加权7 6间平衡会话。M值在增加D激活H待1间D扩9 O会a供|佳D：载y衡。

cp_create_parms.max_tdm_nodes

I存储在拓扑}] b中D最大Z c }（c m > 无限制）。

cp_create_parms.max_tdm_tgs

I存储在拓扑}] b中D最大 TG }（c m > 无限制）。

cp_create_parms.max_isr_sessions

Z c I同1 N k D ISR 会话D最大}。

cp_create_parms.isr_sessions_upper_threshold

见 **cp_create_parms.isr_sessions_lower_threshold**

cp_create_parms.isr_sessions_lower_threshold
P值O限和下限X制Z c D 5塞状，。如果 ISR 会话 } ，过P值O限，则Z c D状，会从非5塞D D d为5塞D。一） ISR 会话 } M Z P值下限， Z c D状，M会 d 回非5塞。

cp_create_parms.isr_max_ru_size
对Z中间会话 y 支V D最大 RU 大小。

cp_create_parms.isr_rcv_pac_window
（议D中间会话D S U w = 窗Z大小。如果Z | Z c ；支V自J & w =，则C值 v C Z中间会话D次级转发O。

cp_create_parms.store_endpt_rscvs
指定G否要为o断而存储 RSCV(AP_YES 或 AP_NO)。

cp_create_parms.store_isr_rscvs
指定G否要为o断而存储 RSCV(AP_YES 或 AP_NO)。

cp_create_parms.store_dlur_rscvs
指定Z c G否要为o断而存储 RSCV(AP_YES 或 AP_NO)。如果C字段h置为AP_YES，则 RSCV 在 QUERY_DLUR_LU 动词中返回。

cp_create_parms.dlur_support
指定C Z c y a供D DLUR D支V级。 | G一v位字段， " I 9 C下P值:

AP_NO
；支V DLUR。

AP_YES
支V DLUR D y P多v子网。

(AP_YES | AP_LIMITED_DLUR_MULTI_SUBNET)
支V \限 DLUR 及 DLUR D多v子网。 v 1 C Z c G端Z c 1， | E P效。

cp_create_parms.pu_conc_support
指定G否支V PU 集中（通#为 AP_NO）。

cp_create_parms.nn_rar
网 g Z c D 7 I = 加阻 9 。

cp_create_parms.hpr_support
指定 I Z c a供D HPR D支V级 (AP_NONE、AP_BASE 或 AP_RTP)。

cp_create_parms.mobile
指定Z c G否G一v HPR 7 6转换X制器(AP_YES 或AP_NO)。如果 **cp_create_parms.hpr_support** 字段未h置为 AP_RTP，则C字段+ # t 。

cp_create_parms.discovery_support
指定C Z c G否9 C²发现²功能。

AP_DISCOVERY_CLIENT
C Z c 9 C²发现²M户机功能

AP_DISCOVERY_SERVER
C Z c 9 C²发现²服务器功能。

cp_create_parms.discovery_group_name

指定C Z c y 9 C D “发现” 功能O 9 C D组 { 。如果C字段 h 置为全 c , 则 9 C缺 ! 组 { 。

cp_create_parms.implicit_lu_0_to_3

指定C Z c G否通过 ACTLU 支V LU 0-3 型D隐 = 定义(AP_YES 或 AP_NO)。

cp_create_parms.default_preference

指定启动C Z c D会话 1 x 行 7 I 选择DW选方 = 。

": 9 C DEFINE_PARTNER_LU 动词以 ? v LU 为基础 2 G | 。

此字段 I 9 C下 P 值:

AP_NATIVE

v 9 C > 机（APPN） 7 I 选择协议。

AP_NONNATIVE

v 9 C非 > 机（AnyNet） 7 I 选择协议。

AP_NATIVE_THEN_NONNATIVE

先 " T > 机（APPN）协议, 如果无法 R = 伙 i LU, 则 9 C非 > 机（AnyNet）协议重 T 会话激活。

AP_NONNATIVE_THEN_NATIVE

先 " T非 > 机（AnyNet）协议, 如果无法 R = 伙 i LU, 则 9 C > 机（APPN）协议重 T 会话激活。

": v 1 Z c Y作员 h) P一 AnyNet DLC, R存在一 v 已定义D AnyNet 4 7 > 1 , 后 f D三 v 值 E P 意义。

cp_create_parms.anynet_supported

指定 AnyNet DLC D支V。C字段 I 以G下 P 值之一

AP_NONE

; 支V任何 ANYNET 功能。C字段 **default_preference** X须 9 C值 AP_NATIVE。

AP_ACCESS_NODE

v 9 C非 > 机（AnyNet） 7 I 选择协议。

AP_NATIVE_THEN_NONNATIVE

C Z c + 支V ANYNET 访问 Z c 功能。

AP_GATEWAY

C Z c + 启动 ANYNET 网关功能。v 1 **node_type** 为 AP_NETWORK_NODE 1 , C值 E G P 效D。

cp_create_parms.comp_in_series

指定G否允许在 RLE 压 u 后 9 C LZ 压 u :

AP_YES

AP_NO

cp_create_parms.max_ls_exception_events

指定 I C Z c 记 < D LS_EXCEPTION 项D最大 } 目。范围从 0 = 200。

cp_create_parms.max_compress_lvl

C Z c 支V D最大压u级。

AP_NONE

C Z c ；支V压u。

AP_RLE_COMPRESSION

C Z c 支V LU 6.2 会话O D RLE 压u和还原，以及#规 LU 会话O D RLE 压u和 LZ9 还原。

AP_LZ9_COMPRESSION

C Z c I 支V LZ9 和 RLE 压u k还原。

AP_LZ10_COMPRESSION

C Z c I 支V LZ10、LZ9 和 RLE 压u k还原。

AP_LZ12_COMPRESSION

C Z c I 支V LZ12、LZ10、LZ9 和 RLE 压u k还原。

cp_create_parms.node_spec_data_len

C字段& < 终h置为c。

cp_create_parms.ptf

C Z配置和X制未来L序暂1修订(PTF)Y作D }组。

cp_create_parms.ptf[0]

REQDISCONT 支V。v人通信或通信服务器通# 9 C REQDISCONT 以M放；再为会话通信？y需D \ 限资源主机4 7。C字Z I C来抑制v人通信或通信服务器9 C REQDISCONT，或修D在I v人通信或通信服务器发M D REQDISCONT k s O 9 C D h置.

AP_SUPPRESS_REQDISCONT

如果h置此位，v人通信或通信服务器；9 C REQDISCONT（忽T C字Z中D y P其|位）。

AP_OVERRIDE_REQDISCONT

如果h置此位，则y]下P = 位，v人通信或通信服务器2 G REQDISCONT O D } # h置。

AP_REQDISCONT_TYPE

如果h置此位，则v人通信或通信服务器在 REQDISCONT O指定类型为“ " 即”。否则，v人通信或通信服务器 指定类型为“ } # ”。（如果未h置 AP_OVERRIDE_REQDISCONT，则忽T此位。）

AP_REQDISCONT_RECONTACT

如果h置此位，则v人通信或通信服务器在 REQDISCONT 中指定“ " 即再*系”。否则，v人通信或通信服务器指定 “； " 即再*系”。（如果未h置 AP_OVERRIDE_REQDISCONT，则忽T此位。）

cp_create_parms.ptf[1]

ERP 支V。

v人通信或通信服务器通# + ACTPU(ERP) 作为 ERP 处理 （ACTPU(ERP) k s 4位 PU-SSCP 会话，+ ；象 ACTPU(cold)，|未k s从t LU-SSCP 和PLU-SLU 会话D隐 = M放）。SNA 5现I以合法X处理 ACTPU(ERP)，F乎|M G ACTPU(cold)。

AP_OVERRIDE_ERP

如果 h 置此位，则 v 人通信或通信服务器 + y P D ACTPU k s 作为 ACTPU(cold)处理。

cp_create_parms.ptf[2]

BIS 支 V。

在M放 \ 限资源 LU 6.2 会话 0， v 人通信或通信服务器一 c 9 C BIS 协议。C字 Z 允许 9 C BIS 来 2 G。

AP_SUPPRESS_BIS

如果 h 置此位，则 v 人通信或通信服务器；9 C BIS 协议。9 C UNBIND（ e ））， " 即M放 \ 限资源 LU 6.2 会话。

up_time

自 Z c 启动（或重新启动）以来D 1 间（以 Y 分之一 k 为%位）。

mem_size

存储器管理从基 > Y 作系统获C D I C存储器D大小。

mem_used

1 0 已分配 x x L D存储器D字 Z }。

mem_warning_threshold

如果，过C分配 P 值，存储器管理 + < G限制存储器资源。

mem_critical_threshold

如果，过C分配 P 值，存储器管理 + < G严 q 限制存储器资源。

nn_functions_supported

t。

functions_supported

指定支V哪些功能。 | I 以G下 P 值中D一 v 或多 v：

AP_NEGOTIABLE_LS
AP_SEGMENT_REASSEMBLY
AP_BIND_REASSEMBLY
AP_PARALLEL_TGS
AP_CALL_IN
AP_ADAPTIVE_PACING
AP_TOPOLOGY_AWARENESS

en_functions_supported

指定支V哪些端 Z c 功能。

AP_SEGMENT_GENERATION

Z c 支V段 z I。

AP_MODE_TO_COS_MAP

Z c 支V至 COS { F 3 d D方 = {。

AP_LOCATE_CDINIT

Z c 支V定位D z I 和定位远L LU D g r D启动 GDS d ?。

AP_REG_WITH_NN

Z c + C Z | 服务网 g Z c 注 a 其 LU。

AP_REG_CHARS_WITH_NN
Z c 支V发M注 a X性（只P 1 支V发M注 a D { F 1 E 支V)。

nn_status
t 。

nn_frsn
t 。

nn_rsn
t 。

def_ls_good_xids
从最后一次启动Z c 起，发 z 在 y P已定义 4 7 > O D I 功D XID ；换D总？。

def_ls_bad_xids
从最后一次启动Z c 起，发 z 在 y P已定义 4 7 > O D ； I 功D XID ；换D总？。

dyn_ls_good_xids
从最后一次启动Z c 起，发 z 在 y P动，4 7 > O D I 功D XID ；换D总？。

dyn_ls_bad_xids
从最后一次启动Z c 起，发 z 在 y P动，4 7 > O D ； I 功D XID ；换D总？。

dlur_release_level
指定 1 0 DLUR D发行级 p 。

nns_dlus_served_lu_reg_supp
v 指端Z c 。指定端Z c D网 g Z c 服务器G否支V I DLUS a 供服务D LU G记。

AP_NO
网 g Z c 服务器；支V I DLUS a 供服务D LU G记。

AP_YES
网 g Z c 服务器支V I DLUS a 供服务D LU G记。

AP_UNKNOWN
端Z c 无网 g Z c 服务器。

v NN: C字段 h 置为 AP_NO。

fq_nn_server_name
1 0网 g Z c 服务器D全限定 { F， $ 度为 17 v 字Z。 | I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构I， R I EBCDIC U q R n d。(? v { F D $ 度最多 I 为 8 v 字Z， " ；含 6 入U q。)

如果C Z c ；G端Z c 或 _ | ；P活动D网 g Z c 服务器，则 + C字段 h 置为 null。

current_isr_sessions

1 0通过C Z c 发M D活动 ISR 会话D } 目。如果C Z c ；G网 g Z c，则 + C字段 h 置为 c 。

nn_functions2

指定G否支V网ｇＺｃ功能。

AP_BRANCH_AWARENESS

ＣＺｃＧ“分支 ”。

branch_ntwk_arch_version

指定支ＶＤ分支网ｇｅ系ａ构Ｄｆ＞，或＿如果ＣＺｃ；支V分支网ｇｅ系ａ构，则为ｃ。

AP_BRANCH_AWARENESS

ＣＺｃＧ“分支 ”。

如果因Ｚｃ Ｐ未启动而９动词；能执行，则Ｌ序返回下ＰＮ}:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而９动词；能执行，则Ｌ序返回下ＰＮ}:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_PARTNER_LU

QUERY_PARTNER_LU 返回 P 关已；> X LU 9 C D伙 i LU D信息。

b 些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关X定伙 i LU D信息，或要获C几 v 『段』中D P m信息，& h 置 **plu_alias** 字段（或 _，如果 **plu_alias** h 置为全 c，则为 **fqplu_name**）。如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST，则忽 T b = v 字段。X须一直 h 置 **lu_name** 或 **lu_alias** 字段。绕却劝 缺 仍孺确染 缺 仍孺缺

```
   unsigned short  act_sess_count;      /* curr active sessions count      */
   unsigned char   partner_cp_name[17]; /* partner LU CP name              */
   unsigned char   partner_lu_located;  /* CP name resolved?               */
} PLU_SUMMARY;

typedef struct plu_detail
{
   unsigned short  overlay_size;        /* size of this entry              */
   unsigned char   plu_alias[8];        /* partner LU alias                */
   unsigned char   fqplu_name[17];      /* fully qualified partner         */
                                        /* LU name                         */
   unsigned char   reserv1;             /* reserved                        */
   unsigned char   description[RD_LEN];
                                        /* resource description            */
   unsigned short  act_sess_count;      /* curr active sessions count      */
   unsigned char   partner_cp_name[17]; /* partner LU CP name              */
   unsigned char   partner_lu_located;  /* CP name resolved?               */
   unsigned char   plu_un_name[8];      /* partner LU uninterpreted name   */
   unsigned char   parallel_sess_supp;  /* parallel sessions supported?    */
   unsigned char   conv_security;       /* conversation security           */
   unsigned short  max_mc_ll_send_size; /* max send LL size for mapped     */
                                        /* conversations                   */
   unsigned char   implicit;            /* implicit or explicit entry      */
   unsigned char   security_details;    /* conversation security detail    */
   unsigned char   duplex_support;      /* full-duplex support             */
   unsigned char   preference;          /* routing preference              */
   unsigned char   reserva[16];         /* reserved                        */
} PLU_DETAIL;
```

&C L序a供下P N }:

opcode

AP_QUERY_PARTNER_LU

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **lu_name** （或_，如果 **lu_name** h置为全c，则为 **lu_alias**）和 **plu_alias**（或_，如果 **plu_alias** h置为全c，则为 **fqplu_name**）D组合（见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c:

AP_FIRST_IN_LIST

忽T **plu_alias** 和 **fqplu_name** 字段，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT
返回 P m从 P m中 4 a 供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE
返回 P m从w引值 y 指定D那项 * <。

lu_name
LU { F。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段 h 置为全 c，则 **lu_alias** 字段 + C Z 确定w引。

lu_alias
> X定义D LU p { 。 | G在> X I 显> 字符集中D一 v 8 字Z字符串。 v 1 **lu_name** 字段 h 置为全 c 1， | E P 效，在 b 种 i v 下。 y P 8 v字Z都G P 效D " R X须 h 置。如果 **lu_name** 和 **lu_alias** 字段都 h 置为全 c，则 9 C k X制 c（缺！ LU）相关 * D LU。

plu_alias
对方 LU D p { 。 | G在> X I 显> 字符集中D一 v 8 字Z字符串。 y P 8 v 字Z都G P 效D， " R X须 h 置。如果C字段 h 置为全 c，则 **fqplu_name** 字段 + 作为w引值 9 C。

fqplu_name
伙 i LU D 17 字Z全限定网 g { F。C { F D $ 度为 17 字Z， I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， R I EBCDIC U q R n d。（？ v { F D $ 度最多 I 为 8 v字Z， " ；含 6 入U q。）

active_sessions
活动会话过K器。指定返回D伙 i LU G否要 y] | G 1 0 P 无活动会话来 x 行 8 选（AP_YES 或 AP_NO）。

返回N数

如果动词 I 功执行，则L序返回下 P N }:

primary_rc
AP_OK

buf_size
在缓 e x 中返回D信息 $ 度。

total_buf_size
返回值 m w返回 y P ； k s D P m信息 y 需要D缓 e x D大小。C值 I 以大Z **buf_size**。

num_entries
5 际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值 I 以大Z **num_entries**。

plu_summary.overlay_size
C项中D字Z }，即 = 下一 v 返回项（如果 P D 话）D偏移 ?。

plu_summary.plu_alias
对方 LU D p { 。 | G在> X I 显> 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P 效D。

plu_summary.fqplu_name

伙 i LU D 17 字 Z 全限定网 g { F。C { F D $ 度为 17 字 Z， I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， R I EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人 U q。）

plu_summary.description

资源 5 w（如在 DEFINE_PARTNER_LU 中 5 w D）。 | G 一 v 以 > X I 显 > 字符集 m > D 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

plu_summary.act_sess_count

> X LU 和伙 i LU 之间 D 活动会话总 }。如果已 + **active_sessions** 过 K 器 h 置为 AP_YES，则 C 字段 + 一直大 Z c。

plu_summary.partner_cp_name

伙 i LU X 制 c D 17 字 Z 全限定网 g { F。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， R I EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人 U q。）

plu_summary.partner_lu_located

指定伙 i LU D X 制 c { F G 否已分 b (AP_YES 或 AP_NO)。

plu_detail.overlay_size

C 项中 D 字 Z }，即 = 下一 v 返回项（如果 P D 话）D 偏移？。

plu_detail.plu_alias

对方 LU D p {。 | G 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。 y P D 8 v 字 Z G P 效 D。

plu_detail.fqplu_name

伙 i LU D 17 字 Z 全限定网 g { F。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， R I EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人 U q。）

plu_detail.description

资源 5 w（如在 DEFINE_PARTNER_LU 中 5 w D）。 | G 一 v 以 > X I 显 > 字符集 m > D 16 字 Z 字符串。 y P 16 v 字 Z 都 P 效。

plu_detail.act_sess_count

> X LU 和伙 i LU 之间 D 活动会话总 }。如果已 + **active_sessions** 过 K 器 h 置为 AP_YES，则 C 字段 + 一直大 Z c。

plu_detail.partner_cp_name

伙 i LU X 制 c D 17 字 Z 全限定网 g { F。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， R I EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v 字 Z， " ；含 6 人 U q。）

plu_detail.partner_lu_located

指定伙 i LU D X 制 c { F G 否已分 b (AP_YES 或 AP_NO)。

plu_detail.plu_un_name

伙 i LU D 非 b M { F。 b G 一 v 8 字 Z A 型 EBCDIC 字符串。

plu_detail.parallel_sess_supp

指定 G 否支 V " 行会话（AP_YES 或 AP_NO）。

plu_detail.conv_security

指定对话 2 全性信息 G 否 I 以发 M 至 C 伙 i LU(AP_YES 或 AP_NO)。如果 h

置为 AP_NO，则B务L序a供D任何2全性信息都；发M至伙 i LU。如果1 0 ；P至C伙 i LU D活动会话，则+ | h置为 AP_UNKNOWN。

plu_detail.max_mc_ll_send_size

I发M至伙 i LU D _ 辑 $ 度(LL)记 < D最大 $ 度。H | 大D }] 记 < 会在发M至伙 i LU 之0中断为几v LL 记 <。最大值 **max_mc_ll_send_size** I以G 32 767。

plu_detail.implict

指定C项G隐 = (AP_YES)定义还G显 = (AP_NO)定义D a 果。

plu_detail.security_details

如同在，S中b v D，返回对话2全性支V。 | I以G下P值中D一v或多v:

AP_CONVERSATION_LEVEL_SECURITY

1从伙 i LU 发v或向伙 i LU 发v分配对话D k s 1， + S \ 对话2全性信息。C下P值h v对话2全性支V D X定类型。

AP_ALREADY_VERIFIED

> X和伙 i LU 都允许S \ 已验证过D分配对话k s。已验证过D k s只需传] C户 ID，而无需传] Z n。

AP_PERSISTENT_VERIFICATION

V C验证在> X和伙 i LU 之间D会话O \ 支V。b M意味着，一) 验证过会话D u < k s（传] C户 ID，一c还P Z n），会话D子序P k s只需传] C户 ID。

AP_PASSWORD_SUBSTITUTION

> X和伙 i LU 支V Z n f 代对话2全性。1发v分配对话k s 1，C k s传] Z n D加 \ 形 =。如果；支V Z n f 代，则C Z n以 e } 文 >（；加 \ ） q = x 行传]。

": 如果C会话；支V Z n f 代，则带P AP_PGM_STRONG 2全类型D ALLOCATE 或 SEND_CONVERSATION + ' \。

AP_UNKNOWN

1 0 ；P至C伙 i LU D活动会话。

plu_detail.duplex_support

如同在，S中b v D，返回通话+工支V。 | G下P值之一:

AP_HALF_DUPLEX

只支V k +工通话。

AP_FULL_DUPLEX

全+工和k +工对话一样支V。

AP_UNKNOWN

因为；P同对方 LU 激活D会话，通话D +工支V ；能x行。

plu_detail.preference

4 U DEFINE_PARTNER_LU 动词中y指定D那样，返回7 I选择协议W选项。

AP_NATIVE

v 9 C > 机（APPN）7 I选择协议。

AP_NONNATIVE

9 C非 > 机（Anynet）协议，如果无法R = 伙 i LU，则9 C非 > 机（AnyNet）协议重T会话激活。

AP_NATIVE_THEN_NONNATIVE

先 " T > 机（APPN）协议，如果无法R = 伙 i LU，则9 C > 机（APPN）协议重T会话激活。

AP_USE_DEFAULT_PREFERENCE

在Z c启动1 9 C定义D缺！W选项。（在 START_NODE O h置 | " R I I QUERY_NODE 重w C。）

注意: v 1 Anynet DLC I C Z L序，R存在一v已定义D Anynet 4 7 > 1，非 > 机7 I选择E P意义。k N v Z 76页D『DEFINE_LS』，以获取 | 多信息。

如果在 START_NODE O a供D字段 **anynet_supported** h置为 AP_NO，则C字段X须9 C值 AP_NATIVE 或 AP_USE_DEFAULT_PREFERENCE。

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PLU_NAME

AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS
AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_PARTNER_LU_DEFINITION

QUERY_PARTNER_LU_DEFINITION 返回 P 关先 0 C DEFINE_PARTNER_LU 动词传] D信息。

b些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关X定伙 i LU D信息，或要获C几 v 『段』中D P m信息，& h置 **plu_alias** 字段（或 _，如果 **plu_alias** h置为全 c，则为 **fqplu_name**）。如果 **plu_alias** 字段为非 c，则 + | C Z 确定w引，"忽T **fqplu_name**。如果 **plu_alias** 字段 h 置为全 c，则 **fqplu_name** C Z 确定w引。如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST，那 4 忽T b = v 字段。（在 b 种 i v 下，如果 h 置K AP_LIST_BY_ALIAS **list_options**，则返回 P m + 4 **plu_alias** 排序，否则 + 4 **fqplu_name** 排序）。k N阅Z 12页D 『 i 询Z c 』，以获取关Z如何 9 C P m q = D 3 0 知 6 。

C P m y] 指定D选项 4 **plu_alias** 或 **fqplu_name** 排序。W先 4 { F $ 度排序，然后对相同 $ 度D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。如果选中K AP_LIST_FROM_NEXT，则返回 P m 4 定义D次序（无 [指定D项G否存在）从下一项 * < 。

注意: C动词 v 返回定义信息。QUERY_PARTNER_LU 动词返回 1 C伙 i LU (" 至Y一 v 会话 1 确定D信息。

VCB a 构

```
typedef struct query_partner_lu_definition
{
   unsigned short  opcode;              /* verb operation code           */
   unsigned char   reserv2;             /* reserved                      */
   unsigned char   format;              /* format                        */
   unsigned short  primary_rc;          /* primary return code           */
   unsigned long   secondary_rc;        /* secondary return code         */
   unsigned char   *buf_ptr;            /* pointer to buffer             */
   unsigned long   buf_size;            /* buffer size                   */
   unsigned long   total_buf_size;      /* total buffer size required    */
   unsigned short  num_entries;         /* number of entries             */
   unsigned short  total_num_entries;   /* total number of entries       */
   unsigned char   list_options;        /* listing options               */
   unsigned char   reserv3;             /* reserved                      */
   unsigned char   plu_alias[8];        /* partner LU alias              */
   unsigned char   fqplu_name[17];      /* fully qualified partner       */
                                        /* LU name                       */
} QUERY_PARTNER_LU_DEFINITION;

typedef struct partner_lu_def_summary
{
   unsigned short  overlay_size;        /* size of this entry            */
   unsigned char   plu_alias[8];        /* partner LU alias              */
   unsigned char   fqplu_name[17];      /* fully qualified partner       */
                                        /* LU name                       */
   unsigned char   description[RD_LEN];
                                        /* resource description          */
} PARTNER_LU_DEF_SUMMARY;

typedef struct partner_lu_def_detail
{
   unsigned short  overlay_size;        /* size of this entry            */
   unsigned char   plu_alias[8];        /* partner LU alias              */
   unsigned char   fqplu_name[17];      /* fully qualified partner       */
```

```
                                        /* LU name                          */
   unsigned char   reserv1;             /* reserved                         */
   PLU_CHARS       plu_chars;           /* partner LU characteristics       */
} PARTNER_LU_DEF_DETAIL;

typedef struct plu_chars
{
   unsigned char   fqplu_name[17];      /* fully qualified partner          */
                                        /* LU name                          */
   unsigned char   plu_alias[8];        /* partner LU alias                 */
   unsigned char   description[RD_LEN]; /* resource description             */
   unsigned char   plu_un_name[8];      /* partner LU uninterpreted name    */
   unsigned char   preference;          /* routing preference               */
   unsigned short  max_mc_ll_send_size;
                                        /* max MC send LL size              */
   unsigned char   conv_security_ver;   /* already_verified accepted        */
   unsigned char   parallel_sess_supp;  /* parallel sessions supported?     */
   unsigned char   reserv2[8];          /* reserved                         */
} PLU_CHARS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_PARTNER_LU_DEFINITION

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ；，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ；要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回*要信息。

AP_DETAIL

返回详细信息。

指定D **plu_alias** （或_，如果 **plu_alias** h置为全c，则为 **fqplu_name**）（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

AP_LIST_BY_ALIAS
返回D P m 4 **plu_alias** 排序。 v 1指定K AP_FIRST_IN_LIST 1，C选项P效。如果指定K AP_LIST_FROM_NEXT 或AP_LIST_INCLUSIVE，那4 P m 3序+取v Z y a供D **plu_alias** 或**fqplu_name** G否作为起< c。

plu_alias
对方 LU D p { 。 | G在> X I 显> 字符集中D一v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须h置。如果C字段h置为全c，则 **fqplu_name** 字段C Z指定y需D伙i LU。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

fqplu_name
伙i LU D 17 字Z全限定网g { F。C { F G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I， " C EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z， " ；含6入U q。）v 1 **plu_alias** 字段h置为全c 1， | E P效。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ； k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**

partner_lu_def_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

partner_lu_def_summary.plu_alias
对方 LU D p { 。 | G在> X I 显> 字符集中D一v 8 字Z字符串。 y P D 8 v字Z G P效D。

partner_lu_def_summary.fqplu_name
伙i LU D 17 字Z全限定网g { F。C { F G I = v C EBCDIC c串* D A 型 EBCDIC 字符串构I， " C EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z， " ；含6入U q。）

partner_lu_def_summary.description
资源5 w（如在 DEFINE_PARTNER_LU 中5 w D）。 | G一v以> X I 显> 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

partner_lu_def_detail.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

partner_lu_def_detail.plu_alias

对方 LU D p { 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P效D。

partner_lu_def_detail.fqplu_name

伙 i LU D 17 字Z全限定网 g { F 。 C { F G I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构I , " C EBCDIC U q R n d。(? v { F D $ 度最多I 为 8 v字Z , " ; 含 6 人U q 。)

partner_lu_def_detail.plu_chars.fqplu_name

伙 i LU D 17 字Z全限定网 g { F 。 C { F G I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构I , " C EBCDIC U q R n d。(? v { F D $ 度最多I 为 8 v字Z , " ; 含 6 人U q 。)

partner_lu_def_detail.plu_chars.plu_alias

对方 LU D p { 。

partner_lu_def_detail.plu_chars.description

资源 5 w (如在 DEFINE_PARTNER_LU 中 5 w D)。 | G一 v 以 > X I 显 > 字符集 m > D 16 字Z字符串。 y P 16 v字Z都P效。

partner_lu_def_detail.plu_chars.plu_un_name

伙 i LU D非 b M { F 。 b G一 v 8 字Z A 型 EBCDIC 字符串。

plu_chars.preference

对至此伙 i LU D会话激活 x 行E选D 7 I 选择协议集。此字段I 9 C下P值:

AP_NATIVE

v 9 C > 机 (APPN) 7 I 选择协议。

AP_NONNATIVE

v 9 C非 > 机 (AnyNet) 7 I 选择协议。

AP_NATIVE_THEN_NONNATIVE

先 " T > 机 (APPN) 协议, 如果无法R = 伙 i LU 则 9 C非 > 机 (AnyNet) 协议重T会话激活。

AP_NONNATIVE_THEN_NATIVE

先 " T非 > 机 (AnyNet) 协议, 如果无法R = 伙 i LU 则 9 C > 机 (APPN) 协议重T会话激活。

AP_USE_DEFAULT_PREFERENCE

在Z c 启动 1 9 C定义D缺 ! W选项。

": v 1 Z c Y作员 h) P一 AnyNet DLC, R存在一 v 已定义D AnyNet 4 7 > 1 , 非 > 机 7 I 选择E P意义。

partner_lu_def_detail.plu_chars.max_mc_ll_send_size

I 发M至伙 i LU D _ 辑 $ 度(LL)记 < D最大 $ 度。 H | 大D }] 记 < 会在发M至伙 i LU 之 0 中断为几 v LL 记 < 。最大值 **max_mc_ll_send_size** I 以G 32 767。

partner_lu_def_detail.plu_chars.conv_security_ver

指定G否Z权伙 i LU 代m > X LU 对 **user_ids** x 行验证, 即伙 i LU G否I 在一 v , S k s 中 h 置已验证指 > 符。

AP_YES
AP_NO

partner_lu_def_detail.plu_chars.parallel_sess_supp
指定G否支V " 行会话（AP_YES 或 AP_NO）。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }：

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_PLU_NAME

AP_INVALID_LIST_OPTION

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }：

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }：

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_PORT

QUERY_PORT 返回关Z Z c D端Z D信息P m。C信息I 『确定 }] 』（执行期间动，Q集D }] ）和『定义 }] 』（DEFINE_PORT O D & C L序y a供D }] ）构I。

b些信息以 * 要q = 或详细信息q = D P m返回。如需P关某v X定端Z D信息，或要获C几v『段』中D P m信息，& h置 **port_name** 字段。否则（如果 **list_options** 字段h置为 AP_FIRST_IN_LIST），忽T C字段。k N阅Z 12页D 『 i 询Z c 』，以获取关Z如何9 C P m q = D 3 0知6。

C P m y] **port_name** 排序。W先4 { F $度排序，然后对相同$度D { F x行 ASCII词d ` 辑排序（ y] IBM D 6611 APPN MIB 定序）。如果选中K AP_LIST_FROM_NEXT，则返回D P m 4定义D次序（无 [指定D项G否存在）从下一项 * <。

要返回D端Z D P m I以 | G y t D DLC { F x行8选。在b种i v下，& h置 **dlc_name** 字段（否则，C字段h置为全c）。

VCB a构

```
typedef struct query_port
{
   unsigned short  opcode;             /* verb operation code            */
   unsigned char   attributes;         /* Verb attributes                */
   unsigned char   format;             /* format                         */
   unsigned short  primary_rc;         /* primary return code            */
   unsigned long   secondary_rc;       /* secondary return code          */
   unsigned char   *buf_ptr;           /* pointer to buffer              */
   unsigned long   buf_size;           /* buffer size                    */
   unsigned long   total_buf_size;     /* total buffer size required     */
   unsigned short  num_entries;        /* number of entries              */
   unsigned short  total_num_entries;  /* total number of entries        */
   unsigned char   list_options;       /* listing options                */
   unsigned char   reserv3;            /* reserved                       */
   unsigned char   port_name[8];       /* port name                      */
   unsigned char   dlc_name[8];        /* DLC name filter                */
} QUERY_PORT;

typedef struct port_summary
{
   unsigned short  overlay_size;       /* size of this entry             */
   unsigned char   port_name[8];       /* port name                      */
   unsigned char   description[RD_LEN];
                                       /* resource description           */
   unsigned char   port_state;         /* port state                     */
   unsigned char   reserv1[1];         /* reserved                       */
   unsigned char   dlc_name[8];        /* name of DLC                    */
} PORT_SUMMARY;

typedef struct port_detail
{
   unsigned short  overlay_size;       /* size of this entry             */
   unsigned char   port_name[8];       /* port name                      */
   unsigned char   reserv1[2];         /* reserved                       */
   PORT_DET_DATA   det_data;           /* determined data                */
   PORT_DEF_DATA   def_data;           /* defined data                   */
} PORT_DETAIL;

typedef struct port_det_data
{
   unsigned char   port_state;         /* port state                     */
   unsigned char   dlc_type;           /* DLC type                       */
```

```
   unsigned char   port_sim_rim;       /* port initialization options     */
   unsigned char   reserv1;            /* reserved                        */
   unsigned short  def_ls_good_xids;   /* number of successful XIDs       */
   unsigned short  def_ls_bad_xids;    /* number of unsuccessful XIDs     */
   unsigned short  dyn_ls_good_xids;   /* successful XIDs on dynamic      */
                                       /* LS count                        */
   unsigned short  dyn_ls_bad_xids;    /* failed XIDs on dynamic          */
   unsigned short  num_implicit_links; /* number of implicit links        */
                                       /* active on this port             */
   unsigned char   neg_ls_supp;        /* are negotiable LSs supported?   */
                                       /* LS count                        */
   unsigned char   abm_ls_supp;        /* are ABM LSs supported?          */
   unsigned long   start_time          /* start time                      */
   unsigned char   reserva[12];        /* reserved                        */
} PORT_DET_DATA;

typedef struct port_def_data
{
   unsigned char   description;        /* resource description            */
   unsigned char   dlc_name[8];        /* DLC name associated with port   */
   unsigned char   port_type;          /* port type                       */
   unsigned char   port_attributes[4]; /* port attributes                 */
   unsigned char   implicit_uplink_to_en;
                                       /* implicit links to EN are uplink */
   unsigned char   reserv3[2];         /* NB_BYTE                         */
   unsigned long   port_number;        /* port number                     */
   unsigned short  max_rcv_btu_size;   /* max receive BTU size            */
   unsigned short  tot_link_act_lim;   /* total link activation limit     */
   unsigned short  inb_link_act_lim;   /* inbound link activation limit   */
   unsigned short  out_link_act_lim;   /* outbound link activation limit  */
   unsigned char   ls_role;            /* initial link station role       */
   unsigned char   retry_flags;        /* conditions for automatic retrys */
                                       /* retries                         */
   unsigned short  max_activation_attempts;
                                       /* how many automatic retries      */
   unsigned short  activation_delay_timer;
                                       /* delay between automatic retries */
   unsigned char   reserv1[10];        /* reserved                        */
   unsigned char   implicit_dspu_template[8];
                                       /* implicit DSPU template          */
   unsigned short  implicit_ls_limit   /* max number of implicit links    */
   unsigned char   reserv2             /* reserved                        */
   unsigned char   implicit_dspu_services;
                                       /* implicit links support DSPUs    */
   unsigned short  implicit_deact_timer;
                                       /* Implicit link HPR link          */
                                       /* deactivation timer              */
   unsigned short  act_xid_exchange_limit;
                                       /* activation XID exchange limit   */
   unsigned short  nonact_xid_exchange_limit;
                                       /* non-act. XID exchange limit     */
   unsigned char   ls_xmit_rcv_cap;    /* LS transmit-rcv capability      */
   unsigned char   max_ifrm_rcvd;      /* max number of I-frames that     */
                                       /* can be received                 */
   unsigned short  target_pacing_count;
                                       /* target pacing count             */
   unsigned short  max_send_btu_size;  /* max send BTU size               */
   LINK_ADDRESS    dlc_data;           /* DLC data                        */
   LINK_ADDRESS    hpr_dlc_data;       /* HPR DLC data                    */
   unsigned char   implicit_cp_cp_sess_support;
                                       /* Implicit links allow CP-CP      */
                                       /* sessions                        */
   unsigned char   implicit_limited_resource;
                                       /* Implicit links are              */
                                       /* limited resource                */
   unsigned char   implicit_hpr_support;
                                       /* Implicit links support HPR      */
   unsigned char   implicit_link_lvl_error;
```

```
                                        /* Implicit links support          */
                                        /* HPR link-level error recovery   */
   unsigned char   retired1;            /* reserved                        */
   TG_DEFINED_CHARS default_tg_chars;   /* Default TG chars                */
   unsigned char   discovery_supported;
                                        /* Discovery function supported?   */
   unsigned short  port_spec_data_len;  /* length of port spec data        */
   unsigned short  link_spec_data_len;  /* length of link spec data        */
} PORT_DEF_DATA;

typedef struct link_address
{
        unsigned short  length;         /* length                          */
        unsigned short  reserve1;       /* reserved                        */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                        /* address                         */
} LINK_ADDRESS;

typedef struct tg_defined_chars
{
        unsigned char   effect_cap;     /* effective capacity              */
        unsigned char   reserve1[5];    /* reserved                        */
        unsigned char   connect_cost;   /* connection cost                 */
        unsigned char   byte_cost;      /* byte cost                       */
        unsigned char   reserve2;       /* reserved                        */
        unsigned char   security;       /* security                        */
        unsigned char   prop_delay;     /* propagation delay               */
        unsigned char   modem_class;    /* modem class                     */
        unsigned char   user_def_parm_1;
                                        /* user_defined parameter 1        */
        unsigned char   user_def_parm_2;
                                        /* user_defined parameter 2        */
        unsigned char   user_def_parm_3;
                                        /* user_defined parameter 3        */
} TG_DEFINED_CHARS;

typedef struct port_spec_data
{
       unsigned char  port_data[SIZEOF_PORT_SPEC_DATA];

}  PORT_SPEC_DATA;

typedef struct link_spec_data
{
       unsigned char  link_data[SIZEOF_LINK_SPEC_DATA];

}  LINK_SPEC_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_PORT

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓 e x D指 k。 & C L序 I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓 e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过b v值。 c值m > 无限制。

list_options

m w在P m信息中& C返回 2 4。

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **port_name**（见下P N }）m > w引值， C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中 4 a供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值 y 指定D那项 * <。

port_name

; i询端Z D { F。 | G在> X I显> 字符集中D一v 8 字Z字符串。 y P 8 v字 Z 都 G P 效D， " R X须 h 置。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽T C字段。

dlc_name

DLC { F过K器。 & C + | h置为全 c 或 _ 一 v 在> X I 显> 字符集中D一v 8 字Z字符串。如果h置K C字段，那 4 只返回 t Z C DLC D端Z。如果C字段 h 置为全 c，则忽T C字段。

返回N数

如果动词 I 功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总 } 。 C值 I 以大Z **num_entries**。

port_summary.overlay_size

C项中D字Z }, 即 = 下一 v 返回项（如果 P D 话）D偏移 ? 。

port_summary.port_name

k 此 4 7 > 关 * D端 Z D { F 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P 效D。

port_summary.description

资源 5 w（如在 DEFINE_PORT 中 5 w D ）。 | G一 v 以 > X I 显 > 字符集 m > D 16 字Z字符串。 y P 16 v字Z都P效。

port_summary.port_state

指定C端Z D 1 0状，。

AP_NOT_ACTIVE
AP_PENDING_ACTIVE
AP_ACTIVE
AP_PENDING_INACTIVE

port_summary.dlc_name

DLC D { F 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P 效D。

port_detail.overlay_size

C项中D字Z }（ | 括任何 **link_spec_data**）, 即 = 下一 v 返回项（如果 P D 话）D偏移 ? 。

port_detail.port_name

k 此 4 7 > 关 * D端 Z D { F 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P 效D。

port_detail.det_data.port_state

指定C端Z D 1 0状，。

AP_NOT_ACTIVE
AP_PENDING_ACTIVE
AP_ACTIVE
AP_PENDING_INACTIVE

port_detail.det_data.dlc_type

DLC D类型。 v 人通信或通信服务器支V下 P 类型:

AP_ANYNET
AP_LLC2
AP_OEM_DLC
AP_SDLC
AP_TWINAX
AP_X25

port_detail.det_data.port_sim_rim

指定G否支V h 置 u < 化方 = (SIM)和 S U u < 化方 = (RIM)(AP_YES or AP_NO)。

port_detail.det_data.def_ls_good_xids

从最后一次启动C端Z起，发z在C端Z O D y P已定义4 7 > O D I 功D XID ；换D总？。

port_detail.det_data.def_ls_bad_xids

从最后一次启动C端Z起，发z在C端Z O D y P已定义4 7 > O D ； I 功D XID ；换D总？。

port_detail.det_data.dyn_ls_good_xids

从最后一次启动C端Z起，发z在C端Z O D y P动，4 7 > O D I 功D XID ；换D总？。

port_detail.det_data.dyn_ls_bad_xids

从最后一次启动C端Z起，发z在C端Z O D y P动，4 7 > O D ； I 功D XID ；换D总？。

port_detail.det_data.num_implicit_links

1 0在C端Z O活动D隐 = 4 7总 } 。 | | 括动，4 7和隐 = 4 7， b些4 7都G y] 发现D 9 C y创 (D。 C端Z O y允许D b些4 7 D } 目 \ = PORT_DEF_DATA D **implicit_ls_limit** 字段D限制。

port_detail.def_data.neg_ls_supp

I 协L 4 7 > D支V，AP_YES 或 AP_NO。

port_detail.det_data.abm_ls_supp

ABM 4 7 > D支V。 | G未知D，直 = DLC 启动为止。

AP_NO
AP_YES
AP_UNKNOWN

port_detail.det_data.start_time

从Z c启动 = C端Z最后一次启动y - 过D 1间， C Y分之k b ？。如果C端Z启动，则C字段中 + 返回c。

port_detail.def_data.description

资源5 w（如在 DEFINE_PORT 中5 w D)。 | G一v以 > X I 显 > 字符集m > D 16 字Z字符串。 y P 16 v字Z都P效。

port_detail.def_data.dlc_name

相关D DLC D { F。 | G在 > X I 显 > 字符集中D一v 8 字Z字符串。 y P D 8 v字Z G P效D。

port_detail.def_data.port_type

指定端Z y 9 C线7 D类型。 C值对& Z下P值之一:

AP_PORT_NONSWITCHED
AP_PORT_SWITCHED
AP_PORT_SATF

port_detail.def_data.port_attributes[0]

b G位字段。 | I 能9 C值 AP_NO 或下P值:

AP_RESOLVE_BY_LINK_ADDRESS

| 指定一种分 f = 达呼P D " T，方法G在9 C P载在S U XID3 O D

CP { F（或Z c ID）之0先9C CONNECT_IN O D 4 7 X址x行分f。}非字段 **port_type** h置为 AP_PORT_SWITCHED，否则此位+；忽T。

port_detail.def_data.implicit_uplink_to_en

v C Z BrNN: 指定如果相Z Z c为端Z c，在此端Z下D隐= 4 7 > G O行4 7还G下行4 7。v 1；存在至相同伙i D 4 7 1 E < G此字段D值，因为1存在b样D 4 7 1，+W先C其来确定4 7类型。

AP_NO

隐= 4 7为下行4 7。

AP_YES

隐= 4 7为O行4 7。

其|Z c类型: C字段h置为 AP_NO。

port_detail.def_data.port_number

端Z号。

port_detail.def_data.max_rcv_btu_size

能S U D最大 BTU 大小。

port_detail.def_data.tot_link_act_lim

4 7激活限制总计。

port_detail.def_data.inb_link_act_lim

入> 4 7激活限制。

port_detail.def_data.out_link_act_lim

v > 4 7激活限制。

port_detail.def_data.ls_role

4 7 > G +。I为I协L（AP_LS_NEG）、主（AP_LS_PRI）或次（AP_LS_SEC）。如果 **implicit_hpr_support** h置为 AP_NO，则此N } # t。

port_detail.def_data.implicit_dspu_template

指定 DSPU 模e（C DEFINE_DSPU_TEMPLATE 动词定义），如果> X Z c要为在此端Z O激活D隐= 4 7 a供 PU 集中，则+此模e C作定义。如果1 4 7激活1，指定D模e；存在（或已达= 5例极限），则激活+ ' \。|G在> X I显>字符集中D一v 8 字Z字符串。y P 8 v字Z都G P效D，" R X须h置。

如果 **def_data.implicit_dspu_services** 字段未h置为AP_PU_CONCENTRATION，则此字段+ # t。

port_detail.def_data.implicit_ls_limit

指定I同1处Z活动状，D隐= 4 7 > D最大}目，|括动，4 7和为Discovery 激活D 4 7。0 值m >无限制，AP_NO_IMPLICIT_LINKS 值m >；允许隐= 4 7。

def_data.implicit.dspu_services

指定> X Z c在此端Z O激活D隐= 4 7中+ a供x下N PU D服务。|h置为下P值之一:

AP_DLUR

> X Z c + 向下N PU a供 DLUR 服务（9C在 DEFINE_DLUR_DEFAULTS 动词中配置D缺！ DLUS）。此h置v1> X Z c G网g Z c 1 P效。

AP_PU_CONCENTRATION

> X Z c + 向下N PU a供 PU 集中（" + 代f I 指定 DSPU 模e在 **def_data.implicit_dspu_template** 字段中指定D定义）。

AP_NONE

> X Z c ；向下N PU a供任何服务。

port_detail.def_data.retry_flags

此字段指定如果在 **def_data.retry_flags** D DEFINE_LS O h置 AP_INHERIT_RETRY j志，C端Z I自动重T激活D u件。| G一v位字段，" I 9C任何下P 4位 OR（或Y作）组合。

AP_RETRY_ON_START

在" T激活1，如果未从远L Z c S U = 响&，则重T 4 7激活。如果在" T激活1，基> 端Z G非活动D，则L序+ " T激活 |。

AP_RETRY_ON_FAILURE

如果在活动和暂挂活动1 4 S ' \，则重T 4 7激活。如果在" T激活1，基> 端Z发z故O，则L序+ " T激活 |。

AP_RETRY_ON_DISCONNECT

如果4 7 I 远L Z c以 } # 方 = 停止，则重T 4 7激活。

AP_DELAY_APPLICATION_RETRIES

I & C L序启动D 4 7激活重T（9C START_LS 或k s = 4 7激活）+ 9C **activation_delay_timer** x行w =。

AP_DELAY_INHERIT_RETRY

} K此字段中D j 志指定D重T u件外，那些在基> 端Z 定义D **retry_flags** 字段中指定D重T u件也 + 9C。

port_detail.def_data.max_activation_attempts

此字段；z z效果，}非在 **def_data.retry_flags** 中D DEFINE_LS 中至Y h置一v j 志， DEFINE_LS O D **def_data.max_activation_attempts** h置为 AP_USE_DEFAULTS。

C字段指定1远L Z c无响&或基> 端Z ；活动1， L序允许重T D次 }。| | 括自动重T和I & C L序 } 动D激活 " T = _。

如果达 = C极限，则；再x行自动重T。C u件I STOP_LS、STOP_PORT、STOP_DLC 或一v I 功激活4位。START_LS 或 OPEN_LU_SSCP_SEC_RQ z z一v % v激活 " T，| 在激活 ' \ 1 ；再重T。

c m > '无限制'。值 AP_USE_DEFAULTS < 致在 DEFINE_PORT O a供D **max_activiation_attempts** D 9C。

ls_detail.def_data.activation_delay_timer

此字段；z z效果，}非在 **def_data.retry_flags** 中D DEFINE_LS 中至Y h置一v j 志， DEFINE_LS O D **def_data.max_activation_attempts** h置为 AP_USE_DEFAULTS。

C字段指定在自动重T之间L序H待D k }，以及如果在 **def_data.retry_flags** 中h置 AP_DELAY_APPLICATION_RETRIES 位， I & C L序 } 动D激活 " T 之间D k }。

值 AP_USE_DEFAULTS < 致在 DEFINE_PORT O a 供D **activiation_delay_timer** D 9 C。

如果指定c，则L序 + 9 C缺！计 1 器 1 间 30 k。

def_data.implicit_dspu_template

指定 DSPU 模e（ C DEFINE_DSPU_TEMPLATE 动词定义），如果 > X Z c 要为在此端Z O激活D隐 = 4 7 a 供 PU 集中，则 + 此模 e C作定义。如果 1 4 7 激活 1，指定D模 e ；存在（或已达 = 5 例极限），则激活 + ' \。 | G 在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须h置。

如果 **def_data.implicit_dspu_services** 字段未h置为 AP_PU_CONCENTRATION，则此字段 + # t 。

def_data.implicit.dspu_services

指定 > X Z c 在此端Z O激活D隐 = 4 7 中 + a 供 x 下N PU D服务。 | h置为下P值之一:

AP_DLUR

> X Z c + 向下N PU a 供 DLUR 服务（ 9 C在 DEFINE_DLUR_DEFAULTS 动词中配置D缺！ DLUS）。

AP_PU_CONCENTRATION

> X Z c + 向下N PU a 供 PU 集中（ " + 代 f I 指定 DSPU 模 e 在 **def_data.implicit_dspu_template** 字段中指定D定义）。

AP_NONE

> X Z c ；向下N PU a 供任何服务。

def_data.implicit_deact_timer

\ 限资源M放计 1 器（ k ）。如果 **implicit_limited_resource** h 置为 AP_YES 或 AP_NO_SESSIONS，则如果在C计 1 器期间； P }] 在 4 7 O穿过， _ P HPR 能 & D隐 = 4 7 + 自动M放， R无会话 9 C 4 7 。

如果 **implicit_limited_resource** h 置为 AP_INACTIVITY，则如果在C计 1 器期间； P }] 在 4 7 O穿过，隐 = 4 7 + 自动M放。

如果指定c，则 + 9 C缺！值 30。否则，最小值为 5。（如果 h 置 | 小D值，则指定D值 + ；忽T， " 9 C 5。）注意， } 非 **implicit_limited_resource** h 置为 AP_NO，否则此N } # t 。

port_detail.def_data.act_xid_exchange_limit

激活 XID ；换限制。

port_detail.def_data.nonact_xid_exchange_limit

非激活 XID ；换限制。

port_detail.def_data.ls_xmit_rcv_cap

指定 4 7 > 发M/ S U能 &。 I 为 + 向同 1（AP_LS_TWS），或 _ 为 + 向； f （AP_LS_TWA）。

port_detail.def_data.max_ifrm_rcvd

> X 4 7 > 在确认发M 0 I S U = D最大 I-帧 } 目。范围: 1--127

port_detail.def_data.target_pacing_count

1 和 32 767 之间D } 字值（ | 括 1 和 32 767），指 > 此 TG O BIND D期望w = 窗Z大小。 v 1 执行固定，S w = 1，此 } 字E P效。 v 人通信或通信服务器1 0 ； 9 C此值。

port_detail.def_data.max_send_btu_size

能发MD最大 BTU 大小。

port_detail.def_data.dlc_data.length

端Z X址 $ 度。

port_detail.def_data.dlc_data.address

端Z X址。

port_detail.def_data.hpr_dlc_data.length

HPR 端Z X址 $ 度。

port_detail.def_data.hpr_dlc_data.address

HPR 端Z X址。目 0 1 支V HPR 4 7 1 9 C。此字段指定I v 人通信或通信服务器发MD信息，存放在9 C此端Z D 4 7 > O ；换 XID3 O X'61' X制向 ? D X'80' 子字段中。

port_detail.def_data.implicit_cp_cp_sess_support

指定对Z在此端Z下D隐 = 4 7 > G否允许 CP-CP 会话（AP_YES 或 AP_NO）。

port_detail.def_data.implicit_limited_resource

指定1 ； P会话9 C 4 7 1， G否M放在此端Z下D隐 = 4 7 >。 | h置为下P值之一:

AP_NO

隐 = 4 7 ； G \ 限资源， R ；会自动M放。

AP_YES or AP_NO_SESSIONS

隐 = 4 7 G \ 限资源， " 1 ； P活动会话9 C | G 1 + 自动M放。

AP_INACTIVITY

隐 = 4 7 G \ 限资源， 1 ； P活动会话9 C | G 1，或1在指定1间内（ I **implicit_deact_timer** 字段指定）4 7 O ； P }] w动1，隐 = 4 7 + 自动M放。

port_detail.def_data.implicit_hpr_support

指定在隐 = 4 7 O G否支V HPR（AP_YES 或 AP_NO）。

port_detail.def_data.implicit_link_lvl_error

指定 HPR 通信 ? G否9 C 4 S c 错误恢4在隐 = 4 7 O发M（AP_YES 或 AP_NO）。

port_detail.def_data.default_tg_chars

TG X性（N阅Z39页D『DEFINE_COS』）。 C Z在此端Z下D隐 = 4 7 >，或指定 **use_default_tg_chars** D已定义4 7 >。

port_detail.def_data.discovery_supported

指定G否在此端Z O执行²发现²Q w功能（AP_YES 或 AP_NO）。

port_detail.def_data.port_spec_data_len

发v ACTIVATE_PORT 信号1，传M = C端Z D未 | D D }] D未n z $ 度（以字Z为%位）。 }] " 置 = PORT_DETAIL a 构中。

port_detail.def_data.link_spec_data_len

在u < 化期间，传M = 4 7 > 组件D未 | D D }] 。 }] " 置 = PORT_DETAIL a 构中， " R t z 在端Z X定D }] 之后。端Z X定D }] k 4 7 X定D }] a 合在一起， + n z 4 字Z范围。在端Z X定D }] k 4 7 X定D }] 之间； P 显 > n d 。

如果因N } 错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PORT_NAME

AP_INVALID_LIST_OPTION

如果因Z c P 未启动而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则L 序返回下 P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_PU

QUERY_PU 返回 > X PU 和 k | G相关D 4 7 D P m。

C信息作为P m返回。如需P关某 v X定 PU D信息，或要获C几 v 『段』中D P m信息，& h 置 **pu_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽T C字段。k N阅Z 12页D 『 i 询Z c 』，以获取关Z如何 9 C P m q = D 3 0 知 6。

C 动词指定G + > X PU k 主机系统直S 相，还G - I DLUR ，S。I + **host_attachment** 字段作为过K器 9 C，以 c v返回P关X定，S类型D信息。

VCB a 构

```
typedef struct query_pu
{
   unsigned short  opcode;            /* verb operation code           */
   unsigned char   attributes;        /* Verb attributes               */
   unsigned char   format;            /* format                        */
   unsigned short  primary_rc;        /* primary return code           */
   unsigned long   secondary_rc;      /* secondary return code         */
   unsigned char   *buf_ptr;          /* pointer to buffer             */
   unsigned long   buf_size;          /* buffer size                   */
   unsigned long   total_buf_size;    /* total buffer size required    */
   unsigned short  num_entries;       /* number of entries             */
   unsigned short  total_num_entries; /* total number of entries       */
   unsigned char   list_options;      /* listing options               */
   unsigned char   reserv3;           /* reserved                      */
   unsigned char   pu_name[8];        /* PU name                       */
   unsigned char   host_attachment;   /* Host Attachment               */
} QUERY_PU;

typedef struct pu_data
{
   unsigned short  overlay_size;      /* size of this entry            */
   unsigned char   pu_name[8];        /* PU name                       */
   unsigned char   description[RD_LEN];
                                      /* resource description          */
   unsigned char   ls_name[8];        /* LS name                       */
   unsigned char   pu_sscp_sess_active;
                                      /* Is PU-SSCP session active     */
   unsigned char   host_attachment;   /* Host attachment               */
   SESSION_STATS   pu_sscp_stats;     /* PU-SSCP session statistics    */
   unsigned char   sscp_id[6];        /* SSCP ID                       */
   unsigned char   conventional_lu_compression;
                                      /* Data compression requested    */
                                      /* for conventional LU sessions  */
   unsigned char   conventional_lu_cryptography;
                                      /* Cryptography required for     */
                                      /* conventional LU sessions      */
   unsigned char   reserva[12];       /* reserved                      */
} PU_DATA;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;       /* session receive RU size       */
   unsigned short  send_ru_size;      /* session send RU size          */
   unsigned short  max_send_btu_size; /* max send BTU size             */
   unsigned short  max_rcv_btu_size;  /* max rcv BTU size              */
   unsigned short  max_send_pac_win;  /* max send pacing window size   */
   unsigned short  cur_send_pac_win;  /* curr send pacing window size  */
   unsigned short  max_rcv_pac_win;   /* max recv pacing window size   */
   unsigned short  cur_rcv_pac_win;   /* current receive pacing        */
```

```
                                    /* window size                    */
   unsigned long   send_data_frames;  /* number of data frames sent    */
   unsigned long   send_fmd_data_frames;
                                    /* num of FMD data frames sent    */
   unsigned long   send_data_bytes;   /* number of data bytes sent     */
   unsigned long   rcv_data_frames;   /* num data frames received      */
   unsigned long   rcv_fmd_data_frames;
                                    /* num of FMD data frames rcvd    */
   unsigned long   rcv_data_bytes;    /* number of data bytes received */
   unsigned char   sidh;              /* session ID high byte          */
                                    /* (from LFSID)                   */
   unsigned char   sidl;              /* session ID low byte           */
                                    /* (from LFSID)                   */
   unsigned char   odai;              /* ODAI bit set                  */
   unsigned char   ls_name[8];        /* Link station name             */
   unsigned char   pacing_type;       /* type of pacing in use         */
} SESSION_STATS;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_PU

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。c值m >无限制。

list_options

m w在P m信息中&C返回2 4。

指定D **pu_name**（见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

pu_name
要P v D Z 一 v PU { 。 b G 一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽 T C 字段。

host_attachment
主机，S D过K器:

AP_NONE
返回 P 关 y P > X PU D信息。

AP_DLUR_ATTACHED
返回 P 关 \ DLUR 支V D y P > X PU D信息。

AP_DIRECT_ATTACHED
v 返回 P 关直 S k 主机系统，S D那些 PU D信息。

返回N数

如果动词 I 功执行，则L序返回下 P N }:

primary_rc
AP_OK

buf_size
在缓 e x 中返回D信息 $ 度。

total_buf_size
返回值 m w返回 y P ； k s D P m信息 y 需要D缓 e x D大小。C值 I 以大Z **buf_size**。

num_entries
5 际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值 I 以大Z **num_entries**。

pu_data.overlay_size
C 项中D字Z }，即 = 下一 v 返回项（如果 P D 话）D偏移 ?。

pu_data.pu_name
PU { F。 b G 一 v 8 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。

pu_data.description
资源 5 w（如在 DEFINE_LS 或 DEFINE_INTERNAL_PU 中 5 w D ）。 | G 一 v 以 > X I 显 > 字符集 m > D 16 字Z字符串。 y P 16 v 字Z都 P 效。

pu_data.ls_name
k 此 PU 相关D 4 7 > D { F。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v 字Z G P 效D。

pu_data.pu_sscp_sess_active
指定 PU-SSCP 会话G否G活动D(AP_YES 或 AP_NO)。

pu_data.host_attachment
> X PU 主机，S 类型:

AP_DLUR_ATTACHED
PU 9 C DLUR 同主机系统，S。

AP_DIRECT_ATTACHED
PU 同主机系统直S ，S。

pu_data.pu_sscp_stats.rcv_ru_size
<终# t C字段。

pu_data.pu_sscp_stats.send_ru_size
<终# t C字段。

pu_data.pu_sscp_stats.max_send_btu_size
能发MD最大 BTU 大小。

pu_data.pu_sscp_stats.max_rcv_btu_size
能SUD最大 BTU 大小。

pu_data.pu_sscp_stats.max_send_pac_win
<终+C字段h置为c。

pu_data.pu_sscp_stats.cur_send_pac_win
<终+C字段h置为c。

pu_data.pu_sscp_stats.max_rcv_pac_win
<终+C字段h置为c。

pu_data.pu_sscp_stats.cur_rcv_pac_win
<终+C字段h置为c。

pu_data.pu_sscp_stats.send_data_frames
发MD}#w}]帧D}?。

pu_data.pu_sscp_stats.send_fmd_data_frames
发MD}#w FMD }]帧D}?。

pu_data.pu_sscp_stats.send_data_bytes
发MD}#}]wD字Z}。

pu_data.pu_sscp_stats.rcv_data_frames
SUD}#w}]帧D}?。

pu_data.pu_sscp_stats.rcv_fmd_data_frames
SUD}#w FMD }]帧D}?。

pu_data.pu_sscp_stats.rcv_data_bytes
SUD}#}]wD字Z}。

pu_data.pu_sscp_stats.sidh
会话 ID _字Z。

pu_data.pu_sscp_stats.sidl
会话 ID M字Z。

pu_data.pu_sscp_stats.odai
原<目DX址指>符。1会话启动1，如果CZc|含主477>，则 ACTPU D发M方置C字段为 0; "R如果 ACTPU G|含从477>DZc，则置C字段为 1。

pu_data.pu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P D 8 v字Z G P效D。

pu_data.pu_sscp_stats.pacing_type
PU-SSCP 会话中 9 C D S U w = 类型。 C N } 能取 AP_NONE b v 值。

sscp_id
b G v | 含从 PU D ACTPU 中S U D SSCP ID D 6 字Z D字段。

如果 **pu_sscp_sess_active** ; G AP_YES, 则C字段置 c 。

pu_data.conventional_lu_compression
指定 9 C此 PU D会话G否 k s x 行 }] 压 u 。

AP_NO
> X Z c ; &对在会话O w动D、 9 C此 PU D }] x 行压 u 或 b 压 u 。

AP_YES
如果主机 k s 对 }] x 行压 u , 则 }] 压 u &对 k 此 PU 关 * D会话启C。

pu_data.conventional_lu_cryptography
指定 k 此 PU 相关D # 规 LU 会话G否要 s 会话 c 加 \ 。

AP_NONE
会话 c 加 \ ; I L 序执行。

AP_MANDATORY
如果 LU P 一 < 入 \ 钥, 则 ? 制性会话 c 加 \ I L 序执行。否则, | X 须 I 9 C LU D & C L 序执行 (如果 b G PU 集中, 则 I 下N LU 执行)。

AP_OPTIONAL
此值允许 I 主机& C L 序以 ? 会话为基础来 } 动加 \ 。如果主机 k s 对一 v k 此 PU 关 * D 会话 x 行加 \ , 则 L 序 D 行为 X w k AP_MANDATORY 相同。如果主机未 k s 加 \ , 则其行为 X w k AP_NONE 相同。

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_PU_NAME

AP_INVALID_PU_TYPE
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_RTP_CONNECTION

在网 g Z c 或端 Z c 处 9 C QUERY_RTP_CONNECTION，返回 P 关 l Y 传 d 协议(RTP)，S（对 Z C，S 来 5， Z c G 一 v 端 c ）D P m 信息。

b 些信息以 * 要 q = 或详细信息 q = D P m 返回。如需 P 关某 v X 定 RTP ，S D 信息，或要获 C 几 v 『段』中 D P m 信息， & h 置 **rtp_name** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C 字段。 k N 阅 Z 12 页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

C P m y] **rtp_name** 排序。W 先 4 { F $ 度排序，然后对相同 $ 度 D { F x 行 ASCII 词 d ` 辑排序（ y] j 准 MIB 定序）。如果选中 K AP_LIST_FROM_NEXT，则返回 P m 4 定义 D 次序（无 [指定 D 项 G 否存在）从下一项 * < 。

VCB a 构

```
typedef struct query_rtp_connection
{
   unsigned short  opcode;             /* verb operation code            */
   unsigned char   reserv2;            /* reserved                       */
   unsigned char   format;             /* format                         */
   unsigned short  primary_rc;         /* Primary return code            */
   unsigned long   secondary_rc;       /* Secondary return code          */
   unsigned char   *buf_ptr;           /* pointer to buffer              */
   unsigned long   buf_size;           /* buffer size                    */
   unsigned long   total_buf_size;     /* total buffer size required     */
   unsigned short  num_entries;        /* number of entries              */
   unsigned short  total_num_entries;  /* total number of entries        */
   unsigned char   list_options;       /* listing options                */
   unsigned char   reserv3;            /* reserved                       */
   unsigned char   rtp_name[8];        /* name of RTP connection         */
} QUERY_RTP_CONNECTION;

typedef struct rtp_connection_summary
{
   unsigned short  overlay_size;       /* size of this entry             */
   unsigned char   rtp_name[8];        /* RTP connection name            */
   unsigned char   first_hop_ls_name[8];
                                       /* LS name of first hop           */
   unsigned char   dest_node_name[17]; /* fully qualified name of        */
                                       /* destination node               */
   unsigned char   reserv1;            /* reserved                       */
   unsigned char   cos_name[8];        /* class-of-service name          */
   unsigned short  num_sess_active;    /* number of active sessions      */
} RTP_CONNECTION_SUMMARY;

typedef struct rtp_connection_detail
{
   unsigned short  overlay_size;       /* size of this entry             */
   unsigned char   rtp_name[8];        /* RTP connection name            */
   unsigned char   first_hop_ls_name[8];
                                       /* LS name of first hop           */
   unsigned char   dest_node_name[17]; /* fully qualified name of        */
                                       /* destination node               */
   unsigned char   isr_boundary_fn;    /* connection provides ISR BF     */
   unsigned char   reserv1[3];         /* reserved                       */
   unsigned char   cos_name[8];        /* class-of-service name          */
   unsigned short  max_btu_size;       /* max BTU size                   */
   unsigned long   liveness_timer;     /* liveness timer                 */
   unsigned char   local_tcid[8];      /* local TCID                     */
   unsigned char   remote_tcid[8];     /* remote TCID                    */
   RTP_STATISTICS  rtp_stats;          /* RTP statistics                 */
```

```
   unsigned short  num_sess_active;    /* number of active sessions      */
   unsigned char   reserv2[16];        /* reserved                       */
   unsigned short  rscv_len;           /* length of appended RSCV        */
} RTP_CONNECTION_DETAIL;

typedef struct rtp_statistics
{
   unsigned long   bytes_sent;         /* total number of bytes sent     */
   unsigned long   bytes_received;     /* total number of bytes received */
   unsigned long   bytes_resent;       /* total number of bytes resent   */
   unsigned long   bytes_discarded;    /* total number bytes discarded   */
   unsigned long   packets_sent;       /* total number of packets sent   */
   unsigned long   packets_received;   /* total number packets received  */
   unsigned long   packets_resent;     /* total number of packets resent */
   unsigned long   packets_discarded;  /* total number packets discarded */
   unsigned long   gaps_detected;      /* gaps detected                  */
   unsigned long   send_rate;          /* current send rate              */
   unsigned long   max_send_rate;      /* maximum send rate              */
   unsigned long   min_send_rate;      /* minimum send rate              */
   unsigned long   receive_rate;       /* current receive rate           */
   unsigned long   max_receive_rate;   /* maximum receive rate           */
   unsigned long   min_receive_rate;   /* minimum receive rate           */
   unsigned long   burst_size;         /* current burst size             */
   unsigned long   up_time;            /* total uptime of connection     */
   unsigned long   smooth_rtt;         /* smoothed round-trip time       */
   unsigned long   last_rtt;           /* last round-trip time           */
   unsigned long   short_req_timer;    /* SHORT_REQ timer duration       */
   unsigned long   short_req_timeouts; /* number of SHORT_REQ timeouts   */
   unsigned long   liveness_timeouts;  /* number of liveness timeouts    */
   unsigned long   in_invalid_sna_frames;
                                       /* number of invalid SNA frames   */
                                       /* received                       */
   unsigned long   in_sc_frames;       /* number of SC frames received   */
   unsigned long   out_sc_frames;      /* number of SC frames sent       */
   unsigned char   reserve[40];        /* reserved                       */
} RTP_STATISTICS;
```

"：C **rtp_connection_detail** 2 G后 z I SNA 定义D 7 I 选择X制向？（RSCV）。在 RTP ，S h 置之后R在任何 7 6 P 换之 0， +存储 RTP ，S D RSCV， " R在？v Z c 处显 > 如下:

- RSCV |含 y P 从 > X Z c = 伙 i RTP Z c D转发。
- 如果伙 i RTP Z c ；G某 v 会话（引起激活 RTP ，S ）D端 c，则 RSCV 还 +存储一 v “_ g 功能转发”， 9 |离 * 伙 i RTP Z c 。
- 即 9 > X Z c ；|含会话端 c， RSCV +；再 | 含引入 > X Z c D _ g 功能转发。

在 7 6 P 换发 z 以后，存储和显 > D RSCV +只 | 含从 > X Z c = 伙 i RTP Z c D转发。（v ；G _ g 功能转发）。

提供N数

&C L 序 a 供下 P N }:

opcode

AP_QUERY_RTP_CONNECTION

format

j 6 VCB D q =。+C字段 h 置为 c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓 e x D指 k。 &C L序 I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL）。

buf_size

y a供D缓 e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大 } ?。项D } ? ；要，过b v值。 c值m > 无限制。

list_options

m w在P m信息中& C返回 2 4。

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

rtp_name m > 一 v w引值， C w引值C Z指定要返回D 5际信息D起 < c。

AP_FIRST_IN_LIST

忽T **rtp_name**，返回 P m从 P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回 P m从 P m中 4 a供D w引值 y 指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回 P m从w引值 y 指定D那项 * <。

rtp_name

RTP ，S { F。 b G v在 I 显 > D > X字符集中D 8 字Z D字符串。 y P 8 v字Z都G P效D， " R X须h置。

返回N数

如果动词 I 功执行，则L序返回下 P N }:

primary_rc

AP_OK

buf_size

在缓 e x中返回D信息 $ 度。

total_buf_size

返回值m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 C值 I 以大Z **buf_size**。

num_entries

5际返回D项D } ?。

total_num_entries

已 - 返回D项D总 }。 C值 I 以大Z **num_entries**。

rtp_connection_summary.overlay_size

C项中D字Z }，即 = 下一 v 返回项（如果P D话）D偏移 ?。

rtp_connection_summary.rtp_name

RTP ，S { F。 b G v在I显> D > X字符集中D 8 字Z D字符串。 y P D 8 v字Z G P效D。

rtp_connection_summary.first_hop_ls_name

RTP ，S D Z一v中继段D 4 7 > { F。 b G v在I显> D > X字符集中D 8 字Z D字符串。 y P D 8 v字Z G P效D。

rtp_connection_summary.dest_node_name

RTP ，S D目D Z c D全限定 17 字Z { F， I = v C EBCDIC c串* D A型 EBCDIC 字符串构I， R I EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）

rtp_connection_summary.cos_name

RTP ，S D服务级{ F。 b G一v 8 字Z字母}字D A 型 EBCDIC 字符串，"以 EBCDIC U q R n d。

rtp_connection_summary.num_sess_active

在 RTP ，S中1 0 P效D会话}。

rtp_connection_detail.overlay_size

C项中D字Z }（|括任何=加 RSCV），即=下一v返回项（如果P D话）D偏移？。

rtp_connection_detail.rtp_name

RTP ，S { F。 b G v在I显> D > X字符集中D 8 字Z D字符串。 y P D 8 v字Z G P效D。

rtp_connection_detail.first_hop_ls_name

RTP ，S D Z一v中继段D 4 7 > { F。 b G v在I显> D > X字符集中D 8 字Z D字符串。 y P D 8 v字Z G P效D。

rtp_connection_detail.dest_node_name

RTP ，S D目D Z c D全限定 17 字Z { F， I = v C EBCDIC c串* D A型 EBCDIC 字符串构I， R I EBCDIC U q R n d。（？v { F D $度最多I为 8 v字Z， " ；含6人U q。）

rtp_connection_detail.isr_boundary_fn

如果 RTP ，S } C Z任一 ISR 会话（> X Z c }向| a供 HPT-APPN _ g功能），则为 AP_YES，否则为 AP_NO。

rtp_connection_detail.cos_name

RTP ，S D服务级{ F。 b G一v 8 字Z字母}字D A 型 EBCDIC 字符串，"以 EBCDIC U q R n d。

rtp_connection_detail.max_btu_size

RTP ，S D最大 btu 大小，以字Z为%位b ？。

rtp_connection_detail.liveness_timer

RTP ，S D z存计1器，以k为%位b ？。

rtp_connection_detail.local_tcid

RTP ，S D > X TCID。

rtp_connection_detail.remote_tcid

RTP ，S D远L TCID。

QUERY_RTP_CONNECTION

rtp_connection_detail.rtp_stats.bytes_sent
C RTP ，S中, ＞XZc已发MD总字Z｝。

rtp_connection_detail.rtp_stats.bytes_received
C RTP ，S中, ＞XZc已SUD总字Z｝。

rtp_connection_detail.rtp_stats.bytes_resent
在转S中, ＞XZc丢'后重发D总字Z｝。

rtp_connection_detail.rtp_stats.bytes_discarded
作为已U=D重4｝]废弃Dm一端 RTP ，S发MD字Z总?。

rtp_connection_detail.rtp_stats.packets_sent
C RTP ，S中＞XZc已发MD(文总?。

rtp_connection_detail.rtp_stats.packets_received
C RTP ，S中＞XZc已U=D(文总?。

rtp_connection_detail.rtp_stats.packets_resent
转发1＞XZc自己丢'后重发D(文总?。

rtp_connection_detail.rtp_stats.packets_discarded
作为已U=D重4｝]废弃Dm一端 RTP ，S发MD(文总?。

rtp_connection_detail.rtp_stats.gaps_detected
＞XZc检b=D间t总?。?一v间t代m着一v或多v丢'D帧。

rtp_connection_detail.rtp_stats.send_rate
C RTP ，S中10D发MYJ（C'位?kb?)。bGC ARB c法计cDy允许D最大发MYJ。

rtp_connection_detail.rtp_stats.max_send_rate
C RTP ，S中最大D发MYJ（C'位?kb?)。

rtp_connection_detail.rtp_stats.min_send_rate
C RTP ，S中最小D发MYJ（C'位?kb?)。

rtp_connection_detail.rtp_stats.receive_rate
C RTP ，S中10DSUYJ（C'位?kb?)。bG在O一次b?间t基础O计cD5际SUYJ。

rtp_connection_detail.rtp_stats.max_receive_rate
C RTP ，S中最大DSUYJ（C'位?kb?)。

rtp_connection_detail.rtp_stats.min_receive_rate
C RTP ，S中最小DSUYJ（C'位?kb?)。

rtp_connection_detail.rtp_stats.burst_size
C RTP ，S中C字Zb?D10突发大小。

rtp_connection_detail.rtp_stats.up_time
C RTP ，S已活动D总Dk｝。

rtp_connection_detail.rtp_stats.smooth_rtt
在＞XZc和对方 RTP Zc之间, 平滑b?D回71间（C毫kb?)。

rtp_connection_detail.rtp_stats.last_rtt
＞XZc和对方 RTP Zc之间最后b?D回71间（C毫kb?)。

rtp_connection_detail.rtp_stats.short_req_timer
C Z SHORT_REQ 定 1 器 D 1 0 V续 1 间（C毫 k b ？）。

rtp_connection_detail.rtp_stats.short_req_timeouts
对 Z C RTP ，S SHORT_REQ 定 1 器已，1 总？。

rtp_connection_detail.rtp_stats.liveness_timeouts
对 Z C RTP ，S 运行着 D 定 1 器已，1 总？。1 ，S 在 **rtp_connection_detail.liveness_timer** 中指定 D 期间中 U 闲，运行着 D 定 1 器，1 。

rtp_connection_detail.rtp_stats.in_invalid_sna_frames
C RTP ，S 中作为无效 D S U 和废弃 D SNA 帧 D 总？。

rtp_connection_detail.rtp_stats.in_sc_frames
C RTP ，S 中 S U D 会话 X 制帧 D 总？。

rtp_connection_detail.rtp_stats.out_sc_frames
C RTP ，S 中发 M D 会话 X 制帧 D 总？。

rtp_connection_detail.num_sess_active
在 RTP ，S 中 1 0 P 效 D 会话 } 。

rtp_connection_detail.rscv_len
RTP ，S D = 加 7 6 选择 X 制向？D $ 度。（如果未 m 加，则 $ 度为 c 。）RSCV + n z 4 字 Z 范围，以确 # } 确对齐下一 v 细 Z 项，+ **rscv_len** ；| 含 C n d 。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_RTP_CONNECTION

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_SESSION

QUERY_SESSION 返回 P 关会话（对 Z b 些会活来 5， Z c M G 端 c ）D P m 信息。

b 些信息以 * 要 q = 或详细信息 q = D P m 返回。如需 P 关某 v X 定会话 D 信息，或要获 C 几 v 『段』中 D P m 信息，& h 置 **session_id** 字段。否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽 T C 字段。注意: X 须 h 置 **lu_name**（或 **lu_alias**）和 **plu_alias**（或 **fqplu_name**）字段。如果 C 字段非 c，则 **lu_name** D 9 C + E 先 Z **lu_alias**。k N 阅 Z 12 页 D 『i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6。

要返回 D 会话 P m I 以 y] 伙 i LU D { F x 行 8 选。要做 = b 一 c， & C h 置 **fqplu_name** 或 **plu_alias** 字段。如果 **plu_alias** h 置为全 c，则 + 9 C **fqplu_name** 值，否则 **plu_alias** + 一直；9 C " R 忽 T **fqplu_name**。

要返回 D 会话 P m I 以 y] k | G 相关 * D 方 = { F x 行 8 选。在 b 种 i v 下，& h 置 **mode_name** 字段（否则，C 字段 h 置为全 c ）。

} K ? v 会话 D 详细信息，如果在 START NODE N } O 指定 C 项，则 +（以 \ 钥 $ 度 q = ）返回 7 I 选择 X 制向 ?（RSVC） 。C RSCV（C *Sna q* = 指定）通过网 g 定义 K 会话以逐段形 = x 行 7 I 。

VCB a 构

```
typedef struct query_session
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   *buf_ptr;          /* pointer to buffer            */
   unsigned long   buf_size;          /* buffer size                  */
   unsigned long   total_buf_size;    /* total buffer size required   */
   unsigned short  num_entries;       /* number of entries            */
   unsigned short  total_num_entries; /* total number of entries      */
   unsigned char   list_options;      /* listing options              */
   unsigned char   reserv3;           /* reserved                     */
   unsigned char   lu_name[8];        /* LU name                      */
   unsigned char   lu_alias[8];       /* LU alias                     */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   fqplu_name[17];    /* fully qualified partner      */
                                      /* LU name                      */
   unsigned char   mode_name[8];      /* mode name                    */
   unsigned char   session_id[8];     /* session ID                   */
} QUERY_SESSION;

typedef struct session_summary
{
   unsigned short  overlay_size;      /* size of this entry           */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   fqplu_name[17];    /* fully qualified partner      */
                                      /* LU name                      */
   unsigned char   reserv3[1];        /* reserved                     */
   unsigned char   mode_name[8];      /* mode name                    */
   unsigned char   session_id[8];     /* session ID                   */
   FQPCID          fqpcid;            /* fully qualified procedure    */
                                      /* correlator ID                */
} SESSION_SUMMARY;
```

```
typedef struct session_detail
{
   unsigned short  overlay_size;      /* size of this entry           */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   fqplu_name[17];    /* fully qualified partner      */
                                      /* LU name                      */
   unsigned char   reserv3[1];        /* reserved                     */
   unsigned char   mode_name[8];      /* mode name                    */
   unsigned char   session_id[8];     /* session ID                   */
   FQPCID          fqpcid;            /* fully qualified procedure    */
                                      /* correlator ID                */
   unsigned char   cos_name[8];       /* Class-of-service name        */
   unsigned char   trans_pri;         /* Transmission priority:       */
   unsigned char   ltd_res;           /* Session spans a limited      */
                                      /* resource                     */
   unsigned char   polarity;          /* Session polarity             */
   unsigned char   contention;        /* Session contention           */
   SESSION_STATS   sess_stats;        /* Session statistics           */
   unsigned char   duplex_support;    /* full-duplex support          */
   unsigned char   sscp_id[6];        /* SSCP ID of host              */
   unsigned char   reserva[20];       /* reserved                     */
   unsigned long   session_start_time;/* start time of the session    */
   unsigned short  session_timeout;   /* session timeout              */
   unsigned char   reservb[7];        /* reserved                     */
   unsigned char   plu_slu_comp_lvl;  /* PLU to SLU compression level */
   unsigned char   slu_plu_comp_lvl;  /* SLU to PLU compression level */
   unsigned char   rscv_len;          /* Length of following RSCV     */
} SESSION_DETAIL;

typedef struct fqpcid
{
   unsigned char   pcid[8];           /* pro correlator identifier    */
   unsigned char   fqcp_name[17];     /* orig's network qualified     */
                                      /* CP name                      */
   unsigned char   reserve3[3];       /* reserved                     */
} FQPCID;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;       /* session receive RU size      */
   unsigned short  send_ru_size;      /* session send RU size         */
   unsigned short  max_send_btu_size; /* Maximum send BTU size        */
   unsigned short  max_rcv_btu_size;  /* Maximum rcv BTU size         */
   unsigned short  max_send_pac_win;  /* Max send pacing window size  */
   unsigned short  cur_send_pac_win;  /* Curr send pacing window size */
   unsigned short  max_rcv_pac_win;   /* Max receive pacing win size  */
   unsigned short  cur_rcv_pac_win;   /* Curr rec pacing window size  */
   unsigned long   send_data_frames;  /* Number of data frames sent   */
   unsigned long   send_fmd_data_frames;
                                      /* num of FMD data frames sent  */
   unsigned long   send_data_bytes;   /* Number of data bytes sent    */
   unsigned long   rcv_data_frames;   /* Num data frames received     */
   unsigned long   rcv_fmd_data_frames;
                                      /* num of FMD data frames recvd */
   unsigned long   rcv_data_bytes;    /* Num data bytes received      */
   unsigned char   sidh;              /* Session ID high byte         */
   unsigned char   sidl;              /* Session ID low byte          */
   unsigned char   odai;              /* ODAI bit set                 */
   unsigned char   ls_name[8];        /* Link station name            */
   unsigned char   pacing_type;       /* type of pacing in use        */
} SESSION_STATS;
```

": C会话细Z 2 G后 z I *Sna q* =定义D 7 I 选择X制向？（RSCV）。C X制向？通过网 g 定义K会话 7 I，"在 BIND O传M。如果 START_NODE 动词O D字段 h 置为 AP_YES，则 | 含 RSCV（以 \ 钥 $ 度 q = ）。对Z START_NODE，**rscv_len** h 置为 c 。

QUERY_SESSION

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_SESSION

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4。

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **lu_name**（或_，如果 **lu_name** h置为全c，则为 **lu_alias**）、**pu_alias**（或_，如果 **plu_alias** h置为全c，则为 **fqplu_name**）、 **mode_name** 和 **session_id** （见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T **session_id**，返回P m从P m D Z一项 * <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项 * <。

lu_name

LU { F。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段h置为全c，则 **lu_alias** 字段+ C Z确定w引。

lu_alias

> X定义D LU p {。| G在> X I显> 字符集中D一v 8 字Z字符串。v 1 **lu_name** 字段h置为全c 1，| E P效，在b种i v下。y P 8 v字Z都G P效D " R X须h置。如果 **lu_name** 和 **lu_alias** 字段都h置为全c，则9 C k X制c（缺！ LU）相关 * D LU。

plu_alias

对方 LU D p {。| G在> X I显> 字符集中D一v 8 字Z字符串。y P 8 v字Z都G P效D， " R X须h置。如果C字段h置为全c，则 **fqplu_name** 字段+ C Z确定w引。

fqplu_name
伙 i LU D 17 字 Z 全限定网 g { F。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I，" C EBCDIC U q R n d。（？v { F D $ 度最多 I 为 8 v 字 Z，" ；含 6 入 U q。）

mode_name
方 = { 过 K 器。| & C h 置为全 c 或为一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d。如果 h 置 K C 字段，那 4 只返回 k C 方 = 相关 D 会话。如果 C 字段 h 置为全 c，则忽 T C 字段。

session_id
会话 D 8 v 字 Z j 6 符。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽 T C 字段。

返回 N 数

如果动词 I 功执行，则 L 序返回下 P N }：

primary_rc
AP_OK

buf_size
在缓 e x 中返回 D 信息 $ 度。

total_buf_size
返回值 m w 返回 y P ； k s D P m 信息 y 需要 D 缓 e x D 大小。C 值 I 以大 Z **buf_size**。

num_entries
5 际返回 D 项 D } ？。

total_num_entries
已 - 返回 D 项 D 总 }。C 值 I 以大 Z **num_entries**。

session_summary.overlay_size
C 项中 D 字 Z }，即 = 下一 v 返回项（如果 P D 话）D 偏移？。

session_summary.plu_alias
对方 LU D p { 。| G 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。y P D 8 v 字 Z G P 效 D。

session_summary.fqplu_name
伙 i LU D 17 字 Z 全限定网 g { F。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I，" C EBCDIC U q R n d。

session_summary.mode_name
方 = { 。b G 一 v 8 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头），" 以 EBCDIC U q R n d。

session_summary.session_id
会话 D 8 v 字 Z j 6 符。

session_summary.fqpcid.pcid
过 L 相关因子 ID。b G v 8 字 Z D . y x 制字符串。

session_summary.fqpcid.fqcp_name
全限定X制ｃＤ｛Ｆ。Ｃ 17 字Ｚ｛ＦＧＩ＝ｖＣ EBCDIC ｃ串＊Ｄ Ａ 型 EBCDIC 字符串构Ｉ，ＲＩ EBCDIC ＵｑＲｎｄ。（？ｖ｛ＦＤ＄度最多Ｉ为 8 ｖ字Ｚ，＂；含６人Ｕｑ。）

session_detail.overlay_size
Ｃ项中Ｄ字Ｚ｝（｜括任何＝加 RSCV），即＝下一ｖ返回项（如果ＰＤ话）Ｄ偏移？。

session_detail.plu_alias
对方 LU Ｄｐ｛。｜Ｇ在＞ＸＩ显＞字符集中Ｄ一ｖ 8 字Ｚ字符串。ｙＰＤ 8 ｖ字ＺＧＰ效Ｄ。

session_detail.fqplu_name
伙ｉ LU Ｄ 17 字Ｚ全限定网ｇ｛Ｆ。Ｃ｛ＦＧＩ＝ｖＣ EBCDIC ｃ串＊Ｄ Ａ 型 EBCDIC 字符串构Ｉ，＂Ｃ EBCDIC ＵｑＲｎｄ。

session_detail.mode_name
方＝｛。ｂＧ一ｖ 8 字Ｚ字母｝字Ｄ Ａ 型 EBCDIC 字符串（以一ｖ字母＊头），＂以 EBCDIC ＵｑＲｎｄ。

session_detail.session_id
会话Ｄ 8 ｖ字Ｚｊ６符。

session_detail.fqpcid.pcid
过Ｌ相关因子 ID。ｂＧｖ 8 字ＺＤ．ｙｘ制字符串。

session_detail.fqpcid.fqcp_name
全限定X制ｃＤ｛Ｆ。Ｃ 17 字Ｚ｛ＦＧＩ＝ｖＣ EBCDIC ｃ串＊Ｄ Ａ 型 EBCDIC 字符串构Ｉ，ＲＩ EBCDIC ＵｑＲｎｄ。（？ｖ｛ＦＤ＄度最多Ｉ为 8 ｖ字Ｚ，＂；含６人Ｕｑ。）

session_detail.cos_name
服务级｛Ｆ。ｂＧ一ｖ 8 字Ｚ字母｝字Ｄ Ａ 型 EBCDIC 字符串（以一ｖ字母＊头），＂以 EBCDIC ＵｑＲｎｄ。

session_detail.trans_pri
传ｄＥ先级。｜ｈ置为下Ｐ值之一:

AP_LOW
AP_MEDIUM
AP_HIGH
AP_NETWORK

session_detail.ltd_res
指定会话Ｇ否９Ｃ＼限制Ｄ资源４７（AP_YES 或 AP_NO）。

session_detail.polarity
指定会话Ｄ极性(AP_PRIMARY 或 AP_SECONDARY)。

session_detail.contention
指定会话ｙＣ极性。｜ｍｗ＞Ｘ LU Ｇ否对Ｃ会话(AP_CONWINNER)Ｄ９ＣＰ²Ｅ先＼ｘ权²或＿｜Ｇ否Ｘ须在９ＣＣ会话 (AP_CONLOSER)之０ｘ行ｋｓ。

session_detail.sess_stats.rcv_ru_size
最大ＳＵ RU 大小。

session_detail.sess_stats.send_ru_size
最大发M RU 大小。

session_detail.sess_stats.max_send_btu_size
能发M D最大 BTU 大小。

session_detail.sess_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

session_detail.sess_stats.max_send_pac_win
会话中发M w = 窗 Z D最大 _ 寸。

session_detail.sess_stats.cur_send_pac_win
会话中发M w = 窗 Z D 1 0大小。

session_detail.sess_stats.max_rcv_pac_win
会话中 S U w = 窗 Z D最大 _ 寸。

session_detail.sess_stats.cur_rcv_pac_win
会话中 S U w = 窗 Z D 1 0大小。

session_detail.sess_stats.send_data_frames
发M D } # w }] 帧D } ?。

session_detail.sess_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

session_detail.sess_stats.send_data_bytes
发M D } # }] w D字 Z }。

session_detail.sess_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

session_detail.sess_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

session_detail.sess_stats.rcv_data_bytes
S U D } # }] w D字 Z }。

session_detail.sess_stats.sidh
会话 ID _ 字 Z。

session_detail.sess_stats.sidl
会话 ID M字 Z。

session_detail.sess_stats.odai
原 < 目 D X址指 > 符。 1 启动会话 1， 如果 > X Z c | 含主 4 7 >， 则 BIND D发M方置C字段为 0。如果 BIND D发M方G | 含从 4 7 > D Z c， 则置C字段为 1。

session_detail.sess_stats.ls_name
同统计信息相关 * D 4 7 > { F。 | G在 > X I 显 > 字符集中D一 v 8 字 Z 字符串。 y P D 8 v字 Z G P 效D。 C字段能 C Z + 会话D统计信息 k 会话 }] w过D 4 7 关 * 起来。

session_detail.sess_stats.pacing_type
C会话中 9 C D S U w = D类型。 | I 以取 AP_NONE、AP_PACING_FIXED 或 AP_PACING_ADAPTIVE H值。

session_detail.duplex_support

如同在，S中b v D，返回通话+工支V。 | G下P值之一:

AP_HALF_DUPLEX

只支V k +工通话。

AP_FULL_DUPLEX

全+工和k +工对话一样支V。还支V加Y }]。

session_detail.sscp_id

对Z关* LU 会话, C字段 | 含着从C Z > X LU 3 d PU D主机D ACTPU中S U D SSCP ID。对Z独" LU 会话, C字段+ h置为全二x制c。

session_detail.session_start_time

从 CP 启动= C会话I为活动y - 过D 1间, C Y分之一k b ?。如果C会话；G全活动D，则1处理i询1， +在C字段中返回c。

session_detail.session_timeout

指定k C会话相关D，1。 | I下P值< v:

k > X LU 相关D LU6.2 ，1
k远L LU 相关D LU6.2 ，1
方= ，1
全V ，1
如果C会话在\限资源4 7 O运行，则为\限资源，1

session_detail.plu_slu_comp_lvl

指定从 PLU 发M= SLU D }] D压u L度。

AP_NONE

；9 C压u。

AP_RLE_COMPRESSION

9 C RLE 压u。

AP_LZ9_COMPRESSION

C Z c I支V LZ9 压u。

AP_LZ10_COMPRESSION

C Z c I支V LZ10 压u。

AP_LZ12_COMPRESSION

C Z c I支V LZ12 压u。

session_detail.slu_plu_comp_lvl

指定从 SLU 发M= PLU D }] D压u L度。

AP_NONE

；9 C压u。

AP_RLE_COMPRESSION

9 C RLE 压u。

AP_LZ9_COMPRESSION

C Z c I支V LZ9 压u。

AP_LZ10_COMPRESSION

C Z c I支V LZ10 压u。

AP_LZ12_COMPRESSION
C Z c I 支V LZ12 压u。

session_detail.rscv_len
；m加= **session_detail** a构D RSCV D $ 度。(如果未m加，则 $ 度为c。)
RSCV + n z 4 字Z范围。

如果因N } 错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_SESSION_ID

AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS
AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_SIGNED_ON_LIST

QUERY_SIGNED_ON_LIST 检w P关C户注 a X定 LU D信息。

I **lu_name** 或 **lu_alias** 指定 > X LU。**Buf_ptr**、**buf_size**、**total_buf_size**、**num_entries**、**total_num_entries** 和 **overlay_size** 对Z QUERY 动词 _ P通 # 意义。

项作为 SIGNED_ON_LIST_ENTRY a构D P m返回， **buf_ptr** 指向C a构，如果 **buf_ptr** 为 NULL，则C a构 = 加在 QUERY_SIGNED_ON_LIST VCB D后 f 。C P m依次 4 **plu_alias/fqplu_name**、**user_id** 和 **profile** 排序。如果已指定 **plu_alias**，则忽T **fqplu_name**。

list_options I 以 9 C值 AP_FIRST_IN_LIST、AP_LIST_FROM_NEXT、AP_LIST_INCLUSIVE。如果 **list_options** 为 AP_FIRST_IN_LIST，则忽T **plu_alias**、**fqplu_name**、**user_id** 和 **profile**。 **list** 指定要从哪v P m返回项（C P m类型X须为 AP_SIGNED_ON_TO_LIST）。

VCB a构

```
typedef struct query_signed_on_list
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   *buf_ptr;          /* pointer to buffer            */
   unsigned long   buf_size;          /* buffer size                  */
   unsigned long   total_buf_size;    /* total buffer size required   */
   unsigned short  num_entries;       /* number of entries            */
   unsigned short  total_num_entries; /* total number of entries      */
   unsigned char   list_options;      /* listing options              */
   unsigned char   reserv3;           /* reserved                     */
   unsigned char   lu_name[8];        /* LU name                      */
   unsigned char   lu_alias[8];       /* LU alias                     */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   fqplu_name[17];    /* fully qualified partner      */
                                      /* LU name                      */
   unsigned char   user_id[10];       /* User ID                      */
   unsigned char   profile[10];       /* Profile                      */
   unsigned char   list;              /* Signed-on list type          */
} QUERY_SIGNED_ON_LIST;

typedef struct signed_on_list_entry
{
   unsigned short  overlay_size;      /* size of this entry           */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   user_id[10];       /* fully qualified partner      */
   unsigned char   profile[10];       /* profile                      */
} SIGNED_ON_LIST_ENTRY;
```

提供N数

&C L序 a 供下 P N }:

opcode

AP_QUERY_SIGNED_ON_LIST

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

buf_ptr

指向C Z写入P m信息D缓 e x D指 k 。 & C L序 I + }] m加 = VCB D末尾（X须 + **buf_ptr** h 置为 NULL）。

buf_size

y a 供D缓 e x D大小。返回D }] + ; ，过 b v 大小。

num_entries

要返回D项D最大 } ? 。项D } ? ；要，过 b v 值。 c 值 m > 无限制。

list_options

m w在P m信息中& C返回 2 4 。

指定D **lu_name**（或 _，如果 **lu_name** h 置为全 c，则为 **lu_alias**）、**pu_alias**（或 _，如果 **plu_alias** h 置为全 c，则为**fqplu_name**）、**user_id** 和 **profile**（见下P N }）m > w引值，C w引值C Z指定要返回D 5 际信息D起 < c 。

AP_FIRST_IN_LIST

忽T **pu_alias**、**fqplu_name**、**user_id** 和 **profile**，返回P m从P m D Z一项 * < 。

AP_LIST_FROM_NEXT

返回P m从P m中 4 a 供D w引值 y 指定D那项D下一项 * < 。

AP_LIST_INCLUSIVE

返回P m从w引值 y 指定D那项 * < 。

lu_name

LU { F 。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段 h 置为全 c，则 **lu_alias** 字段 + C Z确定w引。

lu_alias

> X定义D LU p { 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 v 1 **lu_name** 字段 h 置为全 c 1， | E P效，在 b 种 i v 下。 y P 8 v字Z都G P效D " R X须 h 置。如果 **lu_name** 和 **lu_alias** 字段都 h 置为全 c，则 9 C k X制 c（缺！ LU）相关 * D LU。

plu_alias

对方 LU D p { 。 | G在 > X I 显 > 字符集中D一 v 8 字Z字符串。 y P 8 v字Z都G P效D， " R X须 h 置。如果C字段 h 置为全 c，则 **fqplu_name** 字段 + C Z确定w引。

fqplu_name

伙 i LU D 17 字Z全限定网 g { F 。C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， " C EBCDIC U q R n d 。（? v { F D $ 度最多 I 为 8 v字Z， " ；含 6 入U q 。）

user_id

C户 ID。& C + | h 置为一 v 10 字Z字母 } 字D A 型 EBCDIC 字符串（以一 v 字母 * 头）， " 以 EBCDIC U q R n d 。如果 h 置K C字段，那 4 只返回 k C方 = 相关D会话。如果 **list_options** h 置为 AP_FIRST_IN_LIST，则忽T C字段。

profile

10 字 Z 字母 } 字 D EBCDIC 字符串。注意: L 序 1 0 v 支 V U E 要文件 （10 v EBCDIC U q ）。如果 **list_options** h 置为 AP_FIRST_IN_LIST, 则忽 T C 字段。

list 注 a P m类型。X 须 + | h 置为 AP_SIGNED_ON_TO_LIST。

返回N数

如果动词 I 功执行, 则 L 序返回下 P N } :

primary_rc

AP_OK

buf_size

在缓 e x 中返回 D 信息 $ 度。

total_buf_size

返回值 m w返回 y P ; k s D P m信息 y 需要 D 缓 e x D 大小。C 值 I 以大 Z **buf_size**。

num_entries

5 际返回 D 项 D } ?。

total_num_entries

已 - 返回 D 项 D 总 } 。C 值 I 以大 Z **num_entries**。

signed_on_list_entry.overlay_size

C 项中 D 字 Z } , 即 = 下一 v 返回项（如果 P D 话）D 偏移 ?。

signed_on_list_entry.plu_alias

对方 LU D p { 。 | G 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。y P D 8 v 字 Z G P 效 D。

signed_on_list_entry.user_id

C 户 ID。b G 一 v 10 字 Z 字母 } 字 D A 型 EBCDIC 字符串（以一 v 字母 * 头）, " 以 EBCDIC U q R n d。

signed_on_list_entry.profile

10 字 Z 字母 } 字 D EBCDIC 字符串。注意, L 序 1 0 v 支 V U E 要文件 （10 v EBCDIC U q ）。

如果因 N } 错误而 9 动词 ; 能执行, 则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_ALIAS

AP_INVALID_LU_NAME
AP_INVALID_PLU_NAME
AP_INVALID_USERID
AP_INVALID_PROFILE
AP_INVALID_LIST
AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因 Z c 停止而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

QUERY_STATISTICS

QUERY_STATISTICS i 询 4 7 > 和端 Z 统计值。v 人通信或通信服务器 + C i 询直 S 传] 至 DLC。统计值 D q = 取 v Z DLC 5 现。

VCB a 构

```
typedef struct query_statistics
{
   unsigned short  opcode;          /* verb operation code        */
   unsigned char   reserv2;         /* reserved                   */
   unsigned char   format;          /* format                     */
   unsigned short  primary_rc;      /* primary return code        */
   unsigned long   secondary_rc;    /* secondary return code      */
   unsigned char   name[8];         /* LS or port name            */
   unsigned char   stats_type;      /* LS or port statistics?     */
   unsigned char   table_type;      /* statistics table requested */
   unsigned char   reset_stats;     /* reset the statistics?      */
   unsigned char   dlc_type;        /* type of DLC                */
   unsigned char   statistics[256]; /* current statistics         */
   unsigned char   reserva[20];     /* reserved                   */
} QUERY_STATISTICS;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_QUERY_STATISTICS

format

j 6 VCB D q =。+ C 字段 h 置为 c，以指定 O f P v D VCB D f >。

name 为 4 7 > 或端 Z 定义 D { F（y] **stats_type** N } D h 置）。| G 在 > X I 显 > 字符集中 D 一 v 8 字 Z 字符串。y P 8 v 字 Z 都 G P 效 D，" R X 须 h 置。v 人通信或通信服务器 9 C | + 响 & k } 确 D 4 7 > 或端 Z 相关。

stats_type

统计值 k s D 资源 D 类型。X 须 + | h 置为下 P 值之一:

AP_LS
AP_PORT

table_type

k s D 统计 m D 类型。X 须 + | h 置为下 P 信息类 p 之一:

AP_STATS_TBL

指定统计信息 + ；返回。

AP_ADMIN_TBL

指定管理信息 + ；返回。

AP_OPER_TBL

指定 Y 作信息 + ；返回。？v 类 p D 返回信息 D q = X 定 Z DLC 5 现。

reset_stats

指定 G 否 & C 4 位统计值(AP_YES 或 AP_NO)。

返回N数

如果动词I功执行，则L序返回下PN}:

primary_rc
AP_OK

dlc_type
DLC D类型。C字段D值X定Z DLC 5现。b些值如下:

AP_ANYNET
AP_LLC2
AP_OEM_DLC
AP_SDLC
AP_TWINAX
AP_X25

statistics
4 7 >或端Z D 1 0统计值。

如果因N}错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LINK_NAME

AP_INVALID_PORT_NAME
AP_INVALID_STATS_TYPE
AP_INVALID_TABLE_TYPE

如果因状，错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_LINK_DEACTIVATED

AP_PORT_DEACTIVATED

如果因Z c P未启动而9动词；能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

QUERY_TP

QUERY_TP 返回 P 关 1 0 } I > X LU 9 C D B 务 L 序 D 信息。

C 信息作为 P m返回。如需 P 关某 v X 定 B 务 L 序 D 信息，或要获 C 几 v 『段』中 D P m 信息，& h 置 **tp_name** 字段。如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST，则忽 T C字段。注意: X须一直 h 置 **lu_name** 或 **lu_alias** 字段。如果 **lu_name** 字段为非 c 值，则 + E 先 9 C |，而；G **lu_alias**。k N阅 Z 12页 D 『 i 询 Z c 』，以获取关 Z 如何 9 C P m q = D 3 0 知 6 。

对 Z 相同 $ 度 D { F， 9 C EBCDIC 词 d ` 辑 3 序， + C P m 4 **tp_name** 排序。C 动词返回在 > X LU * < 9 C TP 1 确定 D 信息。QUERY_TP_DEFINITION 动词 v 返回定义信息。

VCB a 构

```
typedef struct query_tp
{
   unsigned short  opcode;              /* Verb operation code          */
   unsigned char   attributes;          /* verb attributes              */
   unsigned char   format;              /* format                       */
   unsigned short  primary_rc;          /* Primary return code          */
   unsigned long   secondary_rc;        /* Secondary return code        */
   unsigned char   *buf_ptr;            /* pointer to buffer            */
   unsigned long   buf_size;            /* buffer size                  */
   unsigned long   total_buf_size;      /* total buffer size required   */
   unsigned short  num_entries;         /* number of entries            */
   unsigned short  total_num_entries;   /* total number of entries      */
   unsigned char   list_options;        /* listing options              */
   unsigned char   reserv3;             /* reserved                     */
   unsigned char   lu_name[8];          /* LU name                      */
   unsigned char   lu_alias[8];         /* LU alias                     */
   unsigned char   tp_name[64];         /* TP name                      */
} QUERY_TP;

typedef struct tp_data
{
   unsigned short  overlay_size;        /* size of this entry           */
   unsigned char   tp_name[64];         /* TP name                      */
   unsigned char   description[RD_LEN];
                                        /* resource description         */
   unsigned short  instance_limit;      /* max instance count           */
   unsigned short  instance_count;      /* current instance count       */
   unsigned short  locally_started_count;
                                        /* locally started instance     */
                                        /* count                        */
   unsigned short  remotely_started_count;
                                        /* remotely started instance    */
                                        /* count                        */
   unsigned char   reserva[20];         /* reserved                     */
} TP_DATA;

typedef struct tp_spec_data
{
        unsigned char pathname[256];    /* path and TP name             */
        unsigned char parameters[64];   /* parameters for TP            */
        unsigned char queued;           /* queued TP (AP_YES)           */
        unsigned char load_type;        /* type of load-DETACHED/CONSOLE */
        unsigned char dynamic_load;     /* dynamic loading of TP enabled */
        unsigned char reserved[5];      /* max size is 120 bytes        */
} TP_SPEC_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_TP

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

|指>在P m信息中&返回2 4: 指定D **lu_name**（或_如果 **lu_name** h置为全c则G **lu_alias**）和 **tp_name** D组合（见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c。

AP_FIRST_IN_LIST

忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT

返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE

返回P m从w引值y指定D那项* <。

lu_name

LU { F。C { F G v 8字Z D A型 EBCDIC 字符串。如果C字段h置为全c，则 **lu_alias** 字段+ C Z确定w引。

lu_alias

>X定义D LU p {。|G在>X I显>字符集中D一v 8字Z字符串。v 1 **lu_name** 字段h置为全c 1，|E P效，在b种i v下。y P 8 v字Z都G P效D " R X须h置。如果 **lu_name** 和 **lu_alias** 字段都h置为c，则+ 9 C k X制c（缺！ LU）相关* D LU。

tp_name

B务L序{ F。|G一v 64 字Z字符串，以U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

QUERY_TP

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**。

tp_data.overlay_size
C项中D字Z }，即 = 下一v返回项（如果P D话）D偏移?。

tp_data.tp_name
B务L序{ F。 | G一v 64 字Z字符串，以U q R n d。

tp_data.instance.description
资源5 w（如在 DEFINE_TP 中5 w D）。 | G一v以 > X I显 > 字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

tp_data.instance_limit
指定B务L序D " 存活动5例D最大 } ?。

tp_data.instance_count
1 0活动D X定B务L序D 5例 } 目。

tp_data.locally_started_count
> X启动D X定B务L序D 5例 } 目（通过发v TP_STARTED 动词D B务L序）。

tp_data.remotely_started_count
已远L启动D X定B务L序D 5例 } 目（通过一v已S U D , S k s）。

tp_chars.tp_data.pathname
指定7 6和B务L序{ F。

tp_chars.tp_data.parameters
指定B务L序D N }。

tp_chars.tp_data.queued
指定G否对B务L序排队。

tp_chars.tp_data.load_type
指定如何装入B务L序。

如果因N }错误而9动词；能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TP_NAME

AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS
AP_INVALID_LIST_OPTION

如果因Z c P未启动而9动词；能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果因系统错误而9动词；能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_TP_DEFINITION

QUERY_TP_DEFINITION 返回以 0 在 DEFINE_TP 动词O传入D信息和关Z v 人通信或通信服务器定义D B 务L 序D信息。

b 些信息以 * 要 q = 或详细信息 q = D P m返回。如需 P 关某 v X 定B 务L 序D信息，或要获C几 v 『段』中D P m信息，& h 置 **tp_name** 字段。

否则（如果 **list_options** 字段 h 置为 AP_FIRST_IN_LIST），忽T C字段。 k N阅Z 12页D『 i 询Z c 』，以获取关Z如何 9 C P m q = D 3 0 知 6 。

9 C EBCDIC 词 d ` 辑 3 序， + C P m 4 **tp_name** 排序。如果选中K AP_LIST_FROM_NEXT，则返回 P m 4 定义D次序（无 [指定D项G否存在）从下一项 * < 。

C动词 v 返回定义信息。QUERY_TP 动词返回在 > X LU * < 9 C B 务L 序 1 确定D信息。

VCB a 构

```
typedef struct query_tp_definition
{
   unsigned short  opcode;             /* Verb operation code           */
   unsigned char   attributes;         /* verb attributes               */
   unsigned char   format;             /* format                        */
   unsigned short  primary_rc;         /* Primary return code           */
   unsigned long   secondary_rc;       /* Secondary return code         */
   unsigned char   *buf_ptr;           /* pointer to buffer             */
   unsigned long   buf_size;           /* buffer size                   */
   unsigned long   total_buf_size;     /* total buffer size required    */
   unsigned short  num_entries;        /* number of entries             */
   unsigned short  total_num_entries;  /* total number of entries       */
   unsigned char   list_options;       /* listing options               */
   unsigned char   reserv3;            /* reserved                      */
   unsigned char   tp_name[64];        /* TP name                       */
} QUERY_TP_DEFINITION;

typedef struct tp_def_summary
{
   unsigned short  overlay_size;       /* size of this entry            */
   unsigned char   tp_name[64];        /* TP name                       */
   unsigned char   description[RD_LEN];
                                       /* resource description          */
} TP_DEF_SUMMARY;

typedef struct tp_def_detail
{
   unsigned short  overlay_size;       /* size of this entry            */
   unsigned char   tp_name[64];        /* TP name                       */
   TP_CHARS        tp_chars;           /* TP characteristics            */
} TP_DEF_DETAIL;

typedef struct tp_chars
{
   unsigned char   description[RD_LEN];
                                       /* resource description          */
   unsigned char   conv_type;          /* conversation type             */
   unsigned char   security_rqd;       /* security support              */
   unsigned char   sync_level;         /* synchronization level support */
   unsigned char   dynamic_load;       /* dynamic load                  */
   unsigned char   enabled;            /* is the TP enabled?            */
   unsigned char   pip_allowed;        /* program initialization        */
```

```
                                    /* parameters supported          */
   unsigned char   duplex_support;  /* duplex supported              */
   unsigned char   reserv3[9];      /* reserved                      */
   unsigned short  tp__instance_limit; /* limit on currently active TP */
                                    /* instances                     */
   unsigned short  incoming_alloc_timeout;
                                    /* incoming allocation timeout   */
   unsigned short  rcv_alloc_timeout;  /* receive allocation timeout    */
   unsigned short  tp_data_len;     /* TP data length                */
   TP_SPEC_DATA    tp_data;         /* TP data                       */
} TP_CHARS;

typedef struct tp_spec_data
{
  unsigned char pathname[256];      /* path and TP name              */
  unsigned char parameters[64];     /* parameters for TP             */
  unsigned char queued;             /* queued TP (AP_YES)            */
  unsigned char load_type;          /* type of load-DETACHED/CONSOLE */
  unsigned char dynamic_load;       /* dynamic loading of TP enabled */
  unsigned char reserved[5];        /* max size is 120 bytes         */
} TP_SPEC_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_TP_DEFINITION

attributes

动词D t性。C字段G一v位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向C Z写入P m信息D缓e x D指k。&C L序I + }] m加= VCB D末尾（X须+ **buf_ptr** h置为 NULL)。

buf_size

y a供D缓e x D大小。返回D }] + ; ，过b v大小。

num_entries

要返回D项D最大} ?。项D } ? ;要，过b v值。 c值m >无限制。

list_options

m w在P m信息中&C返回2 4:

AP_SUMMARY

v返回 * 要信息。

AP_DETAIL

返回详细信息。

指定D **tp_name**（见下P N }）m > w引值，C w引值C Z指定要返回D 5际信息D起< c:

AP_FIRST_IN_LIST
忽T w引值，返回P m从P m D Z一项* <。

AP_LIST_FROM_NEXT
返回P m从P m中4 a供D w引值y指定D那项D下一项* <。

AP_LIST_INCLUSIVE
返回P m从w引值y指定D那项* <。

tp_name
已定义D B务L序D { F。 | G一v 64 字Z字符串，以U q R n d。如果 **list_options** h置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e x中返回D信息$度。

total_buf_size
返回值m w返回y P ; k s D P m信息y需要D缓e x D大小。C值I以大Z **buf_size**。

num_entries
5际返回D项D } ?。

total_num_entries
已 - 返回D项D总 }。C值I以大Z **num_entries**。

tp_def_summary.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

tp_def_summary.tp_name
定义B务L序D { F。 | G一v 64 字Z字符串，以U q R n d。

tp_def_summary.description
资源5 w（如在 DEFINE_TP 中5 w D）。 | G一v以>X I显>字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

tp_def_detail.overlay_size
C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

tp_def_detail.tp_name
定义B务L序D { F。 | G一v 64 字Z字符串，以U q R n d。

tp_def_detail.tp_chars.description
资源5 w（如在 DEFINE_TP 中5 w D）。 | G一v以>X I显>字符集m > D 16 字Z字符串。y P 16 v字Z都P效。

tp_def_detail.tp_chars.conv_type
指定C B务L序支V D对话类型:

AP_BASIC
AP_MAPPED
AP_EITHER

tp_def_detail.tp_chars.security_rqd
指定G否需要对话2全性信息以启动B务L序（AP_NO 或 AP_YES）。

tp_def_detail.tp_chars.sync_level
指定B务L序y支VD同=级:

AP_NONE
B务L序支V同=级“无”(None)。

AP_CONFIRM_SYNC_LEVEL
B务L序支V同=级“确认”(Confirm)。

AP_EITHER
B务L序支V同=级“无”(None)或“确认”(Confirm)。

AP_SYNCPT_REQUIRED
B务L序支V同=级“同= c”(Sync-point)。

AP_SYNCPT_NEGOTIABLE
B务L序支V同=级“无”(None)、“确认”(Confirm)或“同= c”(Sync-point)。

tp_def_detail.tp_chars.dynamic_load
指定B务L序G否I动，装入（AP_YES 或 AP_NO）。

tp_def_detail.tp_chars.enabled
指定B务L序G否I I功，S（AP_YES 或 AP_NO）。缺！为 AP_NO。

tp_def_detail.tp_chars.pip_allowed
指定B务L序G否I S U L序u <化N }（PIP）（AP_YES 或 AP_NO）。

tp_def_detail.tp_chars.duplex_support
指w B务L序G全+工或k +工。

AP_FULL_DUPLEX
指定B务L序G全+工。

AP_HALF_DUPLEX
指定B务L序G k +工。

AP_EITHER_DUPLEX
指定B务L序I为k +工或全+工

tp_def_detail.tp_chars.tp_instance_limit
同1活动D B务L序5例}目D限制。

tp_def_detail.tp_chars.incoming_alloc_timeout
指定一v =达，S +排队H待 RECEIVE_ALLOCATE D k }。 c m >无，1，因此+；确定X挂起。

tp_def_detail.tp_chars.rcv_alloc_timeout
指定 RECEIVE_ALLOCATE 动词+排队H待一v ，S D k }。 c m >无，1，因此+；确定X挂起。

tp_def_detail.tp_chars.tp_data_len
k 5现无关B务L序}] D $度。

tp_def_detail.tp_chars.tp_data
；传] D k 5 现相关 D B 务 L 序 }] 在 DYNAMIC_LOAD_INDICATION O 未 | D。

如果因 N } 错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_TP_NAME

AP_INVALID_LIST_OPTION

如果因 Z c P 未启动而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果因系统错误而 9 动词；能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

第7章 2 全存储动词

> B Z h v K在网 g Z c O发 v D动词。

© Copyright IBM Corp. 1989, 1997, 1998

SAFE_STORE_TOPOLOGY

SAFE_STORE_TOPOLOGY v在网g Z c O 9 C " R如果Z c重新启动, | + 2全X存储能够以后访问D拓扑a构信息。**restore** j志位C Z m w G否对信息x行存储（AP_NO）或访问（AP_YES）。

存储Z c信息作为q = 化P m返回。为K # t P关X定网g Z c D信息或为K # t在} v 『chunks』 中D P m信息, 则 **index** 字段&C h置。

否则（如果 **list_options** 字段置为 AP_FIRST_IN_LIST）, + 忽T C字段。 k N阅 Z 12页D『 i 询Z c 』 , 以获取关Z如何9 C P m q = D 3 0知6。

C P m 4 **index_node_name** 排序。W先4 { 字$度定序, " R然后对相同$度D { 字4 ASCII k词d 3序定序（ y] IBM D 6611 APPN MIB 定序）。其次, P m 4 **index_node_type** } 值排序。如果储存或恢4 TG , 4 **index.tg_dest_node_name**（MIB 定序）定序, 然后4 **index.tg_dest_node_type**（4 }值）定序, R Z三次4 **index.tg_number**（4 }值）定序。

S A F E _ S T O R E _ T O P O L O G Y 动词f代K SFS_ADJACENT_NN、SFS_NN_TOPOLOGY_NODE 和 SFS_NN_TOPOLOGY_TG 动词。| 9 C X制向?来存储网g拓扑a构信息, 如同 | G在拓扑a构中v现D一样, f代K对 | G D转换和来自i询2 G D网g拓扑a构信息。存储和恢4未知D X制向?, " R a供K校验和以阻止; j准D }] 纳入拓扑a构。

VCB a构

```
typedef struct safe_store_topology
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* format                        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned char   buf_ptr;            /* pointer to buffer             */
        unsigned long   buf_size;           /* buffer size                   */
        unsigned long   total_buf_size;     /* total buffer size required    */
                                            /* to hold all information       */
        unsigned short  num_entries;        /* number of entries             */
        unsigned short  total_num_entries;  /* total number of entries       */
        unsigned char   list_options;       /* listing options               */
        unsigned char   restore;            /* store or restore;             */
        unsigned char   resource_types;     /* resource types (nodes, TGs...)*/
        RESOURCE_INDEX  index;              /* resource index                */
        unsigned long   frsn;               /* flow-reduction sequence       */
                                            /* number                        */
        unsigned char   reserv3[16];        /* reserved                      */
      } SAFE_STORE_TOPOLOGY;

typedef struct resource_index
{
        unsigned char   node_name[17];      /* FQ node name                  */
        unsigned char   node_type;          /* node type                     */
        unsigned char   tg_dest_node_name[17];
                                            /* FQ name of TG destination node*/
        unsigned char   tg_dest_node_type;  /* TG destination node type      */
        unsigned char   tg_number;          /* TG number                     */
        unsigned char   reserv1[3];         /* reserved                      */
} RESOURCE_INDEX;
```

```
typedef struct safe_store_data
{
        unsigned short  overlay_size;       /* overalllength of safe         */
                                            /* store data                    */
        unsigned short  sub_overlay_size;   /* offset to first appended      */
                                            /* resource                      */
        RESOURCE_INDEX  index;              /* index of appended resource    */
        unsigned char   checksum[16];       /* reserved                      */
} RESOURCE_INDEX;

typedef struct safe_store_node_data
{
        unsigned short  overlay_size;       /* overalllength of safe         */
                                            /* store data                    */
        unsigned short  sub_overlay_size;   /* offset to first appended      */
        unsigned char   adjacent;           /* is this NNCP and adjacent     */
                                            /* NNCP?                         */
        unsigned char   reserv1;            /* reserved                      */
        unsigned long   last_frsn_sent;     /* last flow reduction sequence  */
                                            /* num sent (if node is adjacent)*/
                                            /* resource                      */
       unsigned long   last_frsn_rcvd;      /* last flow reduction sequence  */
                                            /* num rcvd (if node is adjacent)*/
       unsigned long   frsn;                /* flow reduction sequence number*/
       unsigned short  days_left            /* days left in database         */
       RESOURCE_INDEX  index;               /* index of appended resource    */
} SAFE_STORE_NODE_DATA;

typedef struct safe_store_tg_data
{
        unsigned short  overlay_size;       /* overalllength of safe         */
                                            /* store data                    */
        unsigned short  sub_overlay_size;   /* offset to first appended      */
                                            /* resource                      */
        unsigned long   frsn;               /* flow reduction sequence number*/
        unsigned short  days_left           /* days left in database         */
        unsigned short  vector_len;         /* length of appended vector(s)  */
} SAFE_STORE_TG_DATA;
```

提供N数

当 **restore** 等Z **AP_NO** 时提供N数

&C L序a供下P N }:

opcode

AP_SAFE_STORE_TOPOLOGY

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C & C L序能够m加}] = VCB D末端。

buf_size

a供D缓e大小。返回}] + ；会，过b v大小。

num_entries

返回项D最大} ?。项D } ? + ；会，过b v值。 c值m > 无限制。

list_options

m w在P m信息中& C返回 2 4。 **resource_types** 和 **index** 指定K（ k N阅下P N ｝）m > C Z指定返回D 5际信息启动c D w引值。

AP_FIRST_IN_LIST

忽T w引值 " R返回P m从P m中D Z一v项 * <。

AP_LIST_FROM_NEXT

在项I a供D w引值指定之后，返回P m从P m中D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从I w引值指定D一项 * <。

restore

m w G否& C恢 4（AP_YES）或存储（AP_NO）信息D j 志位。在b种 i v下，置为 AP_NO 。

resource_types

C位字段X制K要存储D拓扑 a 构｝]。在C字段中，任何下P值D组合I以一起做位字段D或运 c :

AP_SFS_NODES

存储拓扑Z c

AP_SFS_ADJ_NODES

存储相Z Z c

AP_SFS_TGS

存储 TG

"： b三v中 j 志位中至Y P一v X须h置。相Z Z c和拓扑Z c在 APPN 中G分离D 5 e，因此头 = 位 j 志位能够以任何组合h置。

index.node_name

来自网 g 拓扑 a 构｝] b D网 g 完{ 7 6 Z c {。 b v {字G I EBCDIC U q R n d D 17 字Z D相Z X制 c {。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， R ；含6人U q。）C字段v对 = APPN Z c D 4 7 G J 1 D " R否则 + 忽T。如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽T C字段。如果 AP_SFS_NODES 和 AP_SFS_ADJ_NODES 都；P在 **resource_types** 中h置，则同样忽T C字段。

index.node_type

Z c D类型。 b v Z c置为下P之一:

AP_NETWORK_NODE
AP_VRN
AP_LEARN_NODE

如果 **node_type** 未知，则 AP_LEARN_NODE X须指定。如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽T C字段。如果 AP_SFS_NODES 和 AP_SFS_ADJ 都；P在 **resource_types** 中h置，则同样忽T C字段。

index.tg_dest_node_name

TG D完{ 7 6目D Z c {。 b v {字G I EBCDIC U q R n d D 17 字Z D相Z X制 c {。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c

，S。（？v{FD$度最多I为8 v字Z， R；含6入Uq。）C字段v对=APPN ZcD47GJ1D"R否则+忽T。如果 **list_options** 置为AP_FIRST_IN_LIST ，则忽TC字段。如果 AP_SFS_NODES 和AP_SFS_ADJ_NODES 都；P在 **resource_types** 中h置，则同样忽TC字段。

index.tg_dest_node_type
C TG D目DZc类型。bvZc置为下P之一:

AP_NETWORK_NODE
AP_VRN

如果 **tg_dest_node_type** 未知，则 AP_LEARN_NODE X须指定。如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽TC字段。如果 AP_SFS_TGS 也；P在 **resource_types** 中h置，则同样忽TC字段。

index.tg_number
同 TG 相关*D号k。如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽TC字段。如果 AP_SFS_TGS 也；P在 **resource_types** 中h置，则同样忽TC字段。

frsn w动u减J 3序号（frsn）。如果为非c值，那4 vP带大Z或HZC值D FRSN D拓扑a构资源返回。

safe_store_data.overlay_size
C项D$度，|括任何nd。如果I能，bG=下一v SAFE_STORE_DATA 2GD偏移?。

safe_store_data.sub_overlay_size
C项D$度，|括任何nd。bG==加 SAFE_STORE_DATA 或SAFE_STORE_TG_DATA D偏移?。1访问=加}]1总9CC字段。

safe_store_data.index
C项Dw引。b种a构能够在f后D SAFE_STORE_TOPOLOGY 动词中a供，CZPvf后项。如果 **dest_tg_name** 置为全二x制c，则I取SAFE_STORE_NODE_DATA 2G。否则，I取 SAFE_STORE_TG_DATA 2G。

safe_store_data.checksum
追加2G和向?D128 位D校验和。如果C校验和及其下P}]p坏，|极PI能检b=；j准"R\xC动词。

safe_store_node_data.overlay_size
C项D$度，|括任何nd。bG==加 SAFE_STORE_DATA 或SAFE_STORE_TG_DATA D偏移?。

safe_store_node_data.sub_overlay_size
C项D$度，|括任何nd。bG==加 SAFE_STORE_DATA 或SAFE_STORE_TG_DATA D偏移?。1访问=加}]1总9CC字段。

safe_store_node_data.adjacent
AP_YES 或 AP_NO 。如果G AP_YES ，则C项同相Z网gZc相关。

safe_store_node_data.last_frsn_sent
如果 **adjacent** 置为 AP_YES，则C字段VP发M=相Z网gZcD最后 FRSN。否则，C字段+置为c。

safe_store_node_data.last_frsn_rcvd
如果 **adjacent** 置为 AP_YES，则C字段V P发M = 相Z网g Z c D最后 FRSN 。否则， C字段 + 置为c 。

safe_store_node_data.frsn
如果C Z c在网g拓扑a构中v现， b M G拓扑a构资源D w ？ u减3序号。否则， C字段 + 置为c 。

safe_store_node_data.days_left
在C Z c # t在拓扑a构 }] b中D l } > } 之0， } 非 | D存在能够确认。 c 意味着无限制。

safe_store_node_data.vector_len
m加D向？ $ 度。 c 意味着； P 向？m加。

safe_store_tg_data.overlay_size
C 项D $ 度， | 括任何n d 。 b G = = 加 SAFE_STORE_DATA 或 SAFE_STORE_TG_DATA D偏移？。

safe_store_tg_data.sub_overlay_size
C项D $ 度， | 括任何n d 。如果存在，则b G = = 加向？D偏移？。 C字段总C Z访问m加向？。

safe_store_tg_data.frsn
C TG D w ？ u减J 3序号。

safe_store_tg_data.days_left
在C Z c # t在拓扑a构 }] b中D l } > } 之0， } 非 | D存在能够确认。 c 意味着无限制。

safe_store_node_data.vector_len
m加D向？ $ 度。 c 意味着； P 向？m加。

返回N数

如果C动词I 功执行，则L 序返回下P N }:

primary_rc
AP_OK

buf_size
在缓e 中返回D信息$ 度。

total_buf_size
返回值5 w K + 会需要C Z返回y P P v D k s 信息D缓e 大小。 I 以 _ Z **buf_size** 。

total_num_entries
I 能已 - 返回项D总 }。 I 以 _ Z **num_entries** 。

num_entries
5 际返回D项D } 目。

如果C动词因N } 错误而未能执行，则L 序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LIST_OPTION

AP_INVALID_NODE
AP_INVALID_RESOURCE_TYPES
AP_INVALID_TG

提供N数

当 restore 等Z AP_YES 时提供N数

&C L序a供下P N }:

opcode

AP_SAFE_STORE_TOPOLOGY

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C&C L序能够m加}] = VCB D末端。

buf_size

a供D缓e大小。返回}] + ；会，过b v大小。

num_entries

返回项D最大} ?。项D } ? + ；会，过b v值。 c值m > 无限制。

restore

指> G否&C恢4（AP_YES）或存储（AP_NO）信息D j 志位。在b种i v下，置为 AP_NO 。

resource_name

网g完{ 7 6资源{。 b v {字G I EBCDIC U q R n d D 17 字Z D相Z X制c {。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（? v { F D $度最多I为 8 v字Z， R ；含6入U q。）如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽T C字段。

resource_type

C位字段X制K要存储D拓扑a构}]。在C字段中，任何下P值D组合I以一起做位字段D或运c:

AP_SFS_NODES
存储拓扑Z c

AP_SFS_ADJ_NODES
存储相Z Z c

AP_SFS_TGS
恢4 TG

": b三v中j志位中至Y P一v X须h置。相Z Z c和拓扑Z c在 APPN 中G分离D 5 e，因此头=位j志位能够以任何组合h置。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_CHECKSUM_FAILED

AP_DATA_CORRUPT
AP_INVALID_RESOURCE_TYPES

如果C动词因 } 确D START_NODE N } ； P发M而未能执行，则L序返回下P N }:

primary_rc
AP_FUNCTION_NOT_SUPPORTED

如果C动词因系统未内 (网 g Z c 支V而未能执行，则L序返回下P N }:

primary_rc
AP_INVALID_VERB

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

SFS_ADJACENT_NN

"：C动词已I SAFE_STORE_TOPOLOGY f代，"Rv为Kk先0f>DL序D兼容性而#t。

SAFE_STORE_TOPOLOGY v在网gZcO9C"R如果Zc重新启动，+2全X存储能够以后访问D拓扑a构信息。 **restore** j志位CZmwG否存储（AP_NO）或访问（AP_YES）信息。

1 **restore** j志位置为 AP_NO 1，SFS_ADJACENT_NN 返回P关相Z网gZcD信息（即，那些 CP-CP 会话活动D、已活动D或已在某些1侯活动D网gZc）。

SFS 信息作为q=化Pm返回。为K#tX定网gZcD信息或为K#t在}v『chunks』中DPm信息，则 **adj_nncp_name** 字段&Ch置。

否则（如果 **list_options** 字段置为 AP_FIRST_IN_LIST），+忽TC字段。kN阅 Z 12页D『i询Zc』，以获取关Z如何9CPmq=D30知6。

CPm4 **adj_nncp_name** 排序。W先4{字$度定序，"R然后对相同$度D{字4 ASCII k词d3序定序（y] IBM D 6611 APPN MIB 定序）。如果选择K AP_LIST_FROM_NEXT，则从下一项*<DPm4定义定序（指定D项存在或；存在）。

VCB a构

```
typedef struct sfs_adjacent_nn
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* format                        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned char   *buf_ptr;           /* pointer to buffer             */
        unsigned long   buf_size;           /* buffer size                   */
        unsigned long   total_buf_size;     /* total buffer size required    */
                                            /* to hold all information       */
        unsigned short  num_entries;        /* number of entries             */
        unsigned short  total_num_entries;  /* total number of entries       */
        unsigned char   list_options;       /* listing options               */
        unsigned char   restore;            /* store or restore;             */
        unsigned char   adj_nncp_name[17];  /* CP name of adj Network Node   */
} SFS_ADJACENT_NN;

typedef struct adj_nncp_data
{
        unsigned short  overlay_size;       /* size of this entry            */
        unsigned char   adj_nncp_name[17];  /* CP name of adj Network Node   */
        unsigned char   cp_cp_sess_status;  /* CP-CP session status          */
        unsigned COUNTER
                        out_of_seq_tdus;    /* out of sequence TDUs          */
        unsigned long   last_frsn_sent;     /* last FSRN sent                */
        unsigned long   last_frsn_rcvd;     /* last FRSN received            */
        unsigned char   reserva[20];        /* reserved                      */
} ADJ_NNCP_DATA;
```

SFS_ADJACENT_NN

提供N数

当 restore 等Z AP_NO 时提供N数

&C L序a供下P N }:

opcode

AP_SFS_ADJACENT_NN

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C&C L序能够m加}] = VCB D末端。

buf_size

a供D缓e大小。返回}] + ；会，过b v大小。

num_entries

返回项D最大} ?。项D } ? + ；会，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4。**resource_types** 和 **index** 指定K（k N阅下P N }）m > C Z指定返回D 5际信息启动c D w引值。

AP_FIRST_IN_LIST

忽T w引值 " R返回P m从P m中D Z一项* <。

AP_LIST_FROM_NEXT

在那项I a供D w引值指定之后，返回P m从P m中D下一项* <。

AP_LIST_INCLUSIVE

返回P m从I w引值指定D一项* <。

restore

m w G否&C恢4（AP_YES）或存储（AP_NO）信息D j志位。在b种i v下，置为 AP_NO 。

adj_nncp_name

完{ 7 6、17 字Z、相Z网g Z c D CP { I = v A 型 EBCDIC 字符串组I，中间以 EBCDIC c ，S， I EBCDIC U q R n d。（? v { F D $度最多I为 8 v字Z， R；含6入U q。）如果 **list_options** 置为 AP_FIRST_IN_LIST，则忽T C字段。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

buf_size

在缓e中返回D信息$度。

total_buf_size
返回值 5 w K + 会需要 C Z 返回 y P P v D k s 信息 D 缓 e 大小。 I 以 _ Z **buf_size** 。

num_entries
5 际返回 D 项 D } 目。

total_num_entries
I 能已 - 返回项 D 总 } 。 I 以 _ Z **num_entries** 。

adj_nncp_data.overlay_size
C 项中 D 字 Z } , 即 = 下一 v 返回项（如果 P D 话）D 偏移 ? 。

adj_nncp_data.adj_nncp_name
I = v A 型 EBCDIC 字符串组 I , 中间以 EBCDIC c , S , R I EBCDIC U q R n d D 17 字 Z D 相 Z 网 g Z c D CP { 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , R ; 含 6 入 U q 。）

adj_nncp_data.cp_cp_sess_status
CP-CP 会话 D 状 , （AP_ACTIVE 或 AP_INACTIVE）。

adj_nncp_data.out_of_seq_tdus
TDU 从 C Z c S U D out_of_sequence D 号 k 。

adj_nncp_data.last_frsn_sent
发 M x C Z c D 最后 w ? u 减 J 3 序号。

adj_nncp_data.last_frsn_rcvd
从 C Z c S U D 最后 w ? u 减 J 3 序号。

如果 C 动词因 N } 错误而未能执行, 则 L 序返回下 P N } :

primary_rc
AP_OK

secondary_rc
AP_INVALID_ADJ_NNCP_NAME

AP_INVALID_LIST_OPTION

提供 N 数

当 restore 等 Z AP_YES 时提供 N 数

& C L 序 a 供下 P N } :

opcode
AP_SFS_ADJACENT_NN

format
j 6 VCB D q = 。 + C 字段 h 置为 c , 以指定 O f P v D VCB D f > 。

buf_ptr
指向能够写入 P m 信息缓 e D 指 k 。在 **buf_ptr** X 须置为 NULL D i v 下, C & C L 序能够 m 加 }] = VCB D 末端。

num_entries
5 际返回 D 项 D } 目。

SFS_ADJACENT_NN

restore

m w G否& C恢 4 （AP_YES）或存储（AP_NO）信息D j 志位。在b 种 i v 下，置为 AP_NO 。

adj_nncp_data.overlay_size

C项中D字Z }，即 = 下一 v 返回项（如果P D话）D偏移？。

adj_nncp_data.adj_nncp_name

I = v A 型 EBCDIC 字符串组I，中间以 EBCDIC c ，S， R I EBCDIC U q R n d D 17 字Z D相Z网 g Z c D CP {。（？v { F D $ 度最多I 为 8 v字Z， R ；含 6 入U q。）

adj_nncp_data.cp_cp_sess_status

1 **restore** 置为 AP_YES 1，忽T C字段。

adj_nncp_data.out_of_seq_tdus

1 **restore** 置为 AP_YES 1，忽T C字段。

adj_nncp_data.last_frsn_sent

发M x C Z c D最后w ？ u 减 J 3 序号。

adj_nncp_data.last_frsn_rcvd

从C Z c S U D最后w ？ u 减 J 3 序号。

返回N数

如果C动词 I 功执行，则L 序返回下P N }:

primary_rc

AP_OK

如果C 动词因Z c P 未启动而未能执行，则L 序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因 } 确D START_NODE N } ； P 发M而未能执行，则L 序返回下P N }:

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果C 动词因系统未内 (网 g Z c 支V 而未能执行，则L 序返回下P N }:

primary_rc

AP_INVALID_VERB

如果C 动词因系统错误而未能执行，则L 序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

SFS_DIRECTORY

} QUERY_DIRECTORY_ENTRY 动词之外, 如果Z c 重新启动, SFS_DIRECTORY 动词允许缓存在网 g Z c O D > X目 < 2 全X储存 " R能够以后访问。 **restore** j 志位C Z m w G否对信息 x 行存储（AP_NO）或访问（AP_YES）。

1 **restore** j 志位置为 AP_YES 1, SFS_DIRECTORY 允许 9 C **directory_entry_summary** 2 G (" 目 < }] b。为K # t X定网 g Z c D P关信息, 或为K # t } v 『chunks』 中D P m信息, 则 **resource_name** 和 **resource_type** 字段& C h 置。

否则（如果 **list_options** 字段置为 AP_FIRST_IN_LIST）, + 忽T C字段。 k N阅 Z 12页D『 i 询Z c 』 , 以获取关Z如何 9 C P m q = D 3 0 知 6 。

在缓存项中D资源信息和 | G D 8 2 以下 P 次序返回:

Z 一 v 网 g Z c
在网 g Z c 分配D Z 一 v LU
在网 g Z c 分配D Z 二 v LU
...
在网 g Z c 分配D Z ... v LU
C网 g Z c 服务D Z 一 v 终端Z c
在终端Z c （1）分配D Z 一 v LU
在终端Z c （1）分配D Z 二 v LU
...
在终端Z c （1）分配D Z ... v LU
...
C网 g Z c 服务D Z ... v 终端Z c
在终端Z c （n）分配D Z 一 v LU
在终端Z c （n）分配D Z 二 v LU
...
Z 二 v 网 g Z c
... H H ..

VCB a 构

```
typedef struct sfs_directory
{
        unsigned short  opcode;             /* verb operation code         */
        unsigned char   reserv2;            /* reserved                    */
        unsigned char   format;             /* format                      */
        unsigned short  primary_rc;         /* primary return code         */
        unsigned long   secondary_rc;       /* secondary return code       */
        unsigned char   *buf_ptr;           /* pointer to buffer           */
        unsigned long   buf_size;           /* buffer size                 */
        unsigned long   total_buf_size;     /* total buffer size required  */
        unsigned short  num_entries;        /* number of entries           */
        unsigned short  total_num_entries;  /* total number of entries     */
        unsigned char   list_options;       /* listing options             */
        unsigned char   restore;            /* store or restore flag       */
        unsigned char   resource_name[17];  /* network qualified res name  */
        unsigned char   reserv3;            /* reserved                    */
        unsigned short  resource_type;      /* Resource type               */
      } SFS_DIRECTORY;
```

SFS_DIRECTORY

```
typedef struct directory_entry_summary
{
        unsigned short  overlay_size;        /* size of this entry          */
        unsigned char   resource_name[17];   /* network qualified res name  */
        unsigned char   reserve1;            /* reserved                    */
        unsigned short  resource_type;       /* Resource type               */
        unsigned short  real_owning_cp_type; /* real owning CP type         */
        unsigned char   real_owning_cp_name[17];
                                             /* real owning CP name         */
        unsigned char   description[RD_LEN]; /* resource description        */
} DIRECTORY_ENTRY_SUMMARY;
```

提供N数

当 restore 等Z AP_NO 时提供N数

&C L序a供下P N }:

opcode

AP_SFS_ADJACENT_NN

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C&C L序能够m加}] = VCB D末端。

buf_size

a供D缓e大小。返回}] + ；会，过b v大小。

num_entries

返回项D最大} ?。项D } ? + ；会，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4。**resource_name** 和 **resource_type** 指定K（k N阅下P N }）m > K C Z指定返回D 5际信息启动c D w引值。

AP_FIRST_IN_LIST

忽T w引值 " R返回P m从P m中D Z一项* <。

AP_LIST_FROM_NEXT

在那项I a供D w引值指定之后，返回P m从P m中D下一项* <。

restore

m w G否&C恢4（AP_YES）或存储（AP_NO）信息D j志位。在b种i v下，置为 AP_NO 。

resource_name

网g完{ 7 6资源{。{ F $度为 17 v字Z， "以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。(? v { F D $度最多I为 8 v字Z， R ；含6入U q。）如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽T C字段。

resource_type

资源类型。k N阅下P之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE

如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽T C字段。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

buf_size
在缓 e 中返回D信息 $ 度。

total_buf_size
返回值 5 w K + 会需要C Z返回 y P P v D k s 信息D缓 e 大小。 I 以 _ Z **buf_size** 。

total_num_entries
I 能已 - 返回项D总 }。 I 以 _ Z **num_entries** 。

num_entries
5 际返回D项D }目。

directory_entry_summary.overlay_size
C项中D字Z }，即 = 下一 v 返回项（如果P D话）D偏移 ? 。

directory_entry_summary.resource_name
网 g 完 { 7 6 资源 { 。 { F $ 度为 17 v 字Z， " 以 EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组I，中间以一 v EBCDIC c ，S。（ ? v { F D $ 度最多I为 8 v 字Z， R ；含 6 入U q 。）如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽T C字段。

directory_entry_summary.resource_type
资源类型。 k N阅下P之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_RES_NAME

AP_INVALID_LIST_OPTIONAP_INVALID_RES_TYPE

directory_entry_summary.real_owning_cp_type
v v NN 和 BrNN： 5 际 5 P D CP 类型。 I 能G下P之一:

AP_NONE
5 际 5 P D CP G + W资源。

AP_ENCP_RESOURCE
5际5 P D CP ；G + W资源 " R G EN。

其 | Z c类型: C字段置为 AP_NONE 。

directory_entry_summary.real_owning_cp_name
v v NN 和 BrNN : 完{ 7 6 5际5 P D CP {。{ F $度为 17 v字Z, "以 EBCDIC U q R n d。 | I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构I。(? v { F D $度最多I为 8 v字Z, R ；含6入U q。)

如果5际5 P D CP G 8 2, 则C字段置为二x制c。

如果5际5 P D CP ；G 8 2, 那4 C字段置为5际5 P D CP {。

如果在 BrNN D r中资源I EN 5 P, 则在 BrNN D NNS D目<中5际5 P D CP ；G 8 2。在b种i v下, 5际5 P D CP G EN, + 8 2 G BrNN 。

其 | Z c类型: C字段置为二x制c。

提供N数

当 restore 等Z AP_YES 时提供N数

& C L序a供下P N }:

opcode
AP_SFS_DIRECTORY

format
j 6 VCB D q =。 + C字段h置为c, 以指定O f P v D VCB D f >。

buf_ptr
指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下, C & C L序能够m加 }] = VCB D末端。

buf_size
a供D缓e大小。

restore
m w G否& C恢4 (AP_YES) 或存储 (AP_NO) 信息D j 志位。在b种i v下, 置为 AP_NO 。

resource_name
网g完{ 7 6资源{。{ F $度为 17 v字Z, "以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I, 中间以一v EBCDIC c , S。(? v { F D $度最多I为 8 v字Z, R ；含6入U q。)如果C & C L序恢4目< D Z一v 『chunk』 , 那4 b & C置为全c。否则, C & C L序& C在0一v 『chunk』 中h置b v为最后项目D资源{。

resource_type
资源类型。 k N阅下P之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE

如果C & C L 序恢 4 目 < D Z 一 v 『chunk』，那 4 b & C 置为全 c 。

directory_entry_summary.overlay_size

C 项中D字 Z }，即 = 下一 v 返回项（如果 P D 话）D 偏移？。 1 **restore** 置为 AP_NO 1， b X 须同返回值 **overlay_size** 一样。

directory_entry_summary.resource_name

网 g 完 { 7 6 资源 {。{ F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（？v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入U q。）

directory_entry_summary.resource_type

资源类型。 k N阅下 P 之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE

directory_entry_summary.real_owning_cp_type

v v NN 和 BrNN： 5 际 5 P D CP 类型。 I 能G下 P 之一:

AP_NONE

5 际 5 P D CP G + W资源。

AP_ENCP_RESOURCE

5 际 5 P D CP ；G + W资源 " R G EN。

其 | Z c 类型: C字段置为 AP_NONE 。

directory_entry_summary.real_owning_cp_name

v v NN 和 BrNN： 完 { 7 6 5 际 5 P D CP {。{ F $ 度为 17 v 字 Z，" 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（？v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入U q。）

如果 5 际 5 P D CP G 8 2，则C字段置为二 x 制 c 。

如果 5 际 5 P D CP ；G 8 2，那 4 C字段置为 5 际 5 P D CP {。

如果在 BrNN D r 中资源 I EN 5 P，则在 BrNN D NNS D目 < 中 5 际 5 P D CP ；G 8 2。在 b 种 i v 下， 5 际 5 P D CP G EN， + 8 2 G BrNN 。

其 | Z c 类型: C字段置为二 x 制 c 。

返回N数

如果C动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

如果C动词因 N } 错误而未能执行，则 L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_RES_NAME

AP_INVALID_LIST_OPTION

如果C动词因}确D START_NODE N}；P发M而未能执行，则L序返回下P N}:

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果C动词因系统未内(网g Z c支V而未能执行，则L序返回下P N}:

primary_rc

AP_INVALID_VERB

如果C动词因Z c P未启动而未能执行，则L序返回下P N}:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因系统错误而未能执行，则L序返回下P N}:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

SFS_NN_TOPOLOGY_NODE

"：C动词已 I SAFE_STORE_TOPOLOGY f代，" R v为K k先0 f > D L序D兼容性而# t。

？一v网g Z c维护着V P关Z y P网g Z c、VRN 和网g Z c =网g中D网g Z c TG 信息D网g Z c 拓扑a构}] b。如果Z c重新启动，SFS_NN_TOPOLOGY_NODE 动词C Z 2全X储存能够以后访问D拓扑a构}] b Z c项。**restore** j志位C Z m w G否对信息x行存储（AP_NO）或访问（AP_YES）。

为K # t P关X定网g Z c D信息或为K # t } v 『chunks』 中D P m信息，则 **node_name** 和 **node_type** 字段& C h置。

否则（如果 **list_options** 字段置为 AP_FIRST_IN_LIST），+忽T C字段。k N阅 Z 12页D『i询Z c』，以获取关Z如何9 C P m q = D 3 0知6。

C P m 4 U **node_name**、和 **node_name_type**、和 **frsn** 。W先4 {字$度定序，" R然后对相同$度D {字4 ASCII k词d 3序定序（y] IBM D 6611 APPN MIB 定序）。**node_type** 4 AP_NETWORK_NODE 排序，然后4 AP_VRN 排序。**frsn** 4 }字排序。

- 如果选择K AP_LIST_INCLUSIVE，则返回P m从那v {字D Z一v P效记< * <。
- 如果选择K AP_LIST_FROM_NEXT，则C P m +从I C指定{ D Z一v P效记< * <。

注意如果 **frsn** 字段置为非c值，则v P _ Z C值D带P w ? u减3序号（FRSN）D }] b d入返回。b样允许在 『chunks』 D一v号k通过W次C = C Z c D 1 0 FRSN 1，返回一致D拓扑a构}] b。工作如下:

1. 发v返回Z c 1 0 FRSN D QUERY_NODE 。
2. 为C = 『chunks』 中D y P }] b项，！I能D发v许多SFS_NN_TOPOLOGY_NODE（FRSN 置为c）。
3. 再次发v QUERY_NODE " R H O新D FRSN k在Z一W段返回D FRSN。
4. 如果= v FRSN ；同，那4在}] b中y D d D FRSN，发v带P置为大Z Z一W段a供D FRSN D FRSN D SFS_NN_TOPOLOGY_NODE 。

VCB a构

```
typedef struct sfs_nn_topology_node
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* format                        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned char   *buf_ptr;           /* pointer to buffer             */
        unsigned long   buf_size;           /* buffer size                   */
        unsigned long   total_buf_size;     /* total buffer size required    */
        unsigned short  num_entries;        /* number of entries             */
        unsigned short  total_num_entries;  /* total number of entries       */
        unsigned char   list_options;       /* listing options               */
        unsigned char   restore;            /* store or restore;             */
        unsigned char   node_name[17];      /* network qualified            */
                                            /* node name                     */
```

```
        unsigned char   node_type;          /* node type                    */
        unsigned long   frsn;               /* flow-reduction sequence      */
                                            /* number                       */
} SFS_NN_TOPOLOGY_NODE;

typedef struct nn_topology_node_detail
{
        unsigned short  overlay_size;       /* size of this entry           */
        unsigned char   node_name[17];      /* network qualified            */
        unsigned char   node_type;          /* node type                    */
        unsigned short  days_left           /* days left in database        */
        unsigned long   frsn;               /* flow reduction sequence number*/
        unsigned long   rsn;                /* resource sequence number     */
        unsigned char   rar;                /* route additional resistence  */
        unsigned char   status;             /* node status                  */
        unsigned char   function_support;   /* function support             */
        unsigned char   reserv2;            /* reserved                     */
        unsigned char   reserva[20];        /* reserved                     */
} NN_TOPOLOGY_NODE_DETAIL;
```

提供N数

当 **restore** 等Z **AP_NO** 时提供N数

&C L序a供下P N }:

opcode

AP_SFS_NN_TOPOLOGY_NODE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C&C L序能够m加 }] = VCB D末端。

buf_size

a供D缓e大小。返回 }] + ；会，过b v大小。

num_entries

返回项D最大 } ?。项D } ? + ；会，过b v值。 c值m > 无限制。

list_options

m w在P m信息中&C返回2 4。**node_name**、**node_types** 和 **frsn** 指定K（k N阅下P N }）m > K C Z指定返回D 5际信息启动c D w引值。

AP_FIRST_IN_LIST

忽T w引值 " R返回P m从P m中D Z一项 * <。

AP_LIST_FROM_NEXT

在那项I a供D w引值指定之后，返回P m从P m中D下一项 * <。

AP_LIST_INCLUSIVE

返回P m从I w引值指定D一项 * <。

restore

m w G否&C恢4（AP_YES）或存储（AP_NO）信息D j志位。在b种i v下，置为 AP_NO 。

node_name

来自网g拓扑a构 }] b D网g完{ 7 6 Z c {。 b v {字G I EBCDIC U q

R n d D 17 字 Z D 相 Z X 制 c { 。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（？v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q 。）C 字段 v 对 = APPN Z c D 4 7 G J 1 D " R 否则 + 忽 T 。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

node_type

Z c D 类型。 b v Z c 置为下 P 之一:

AP_NETWORK_NODE
AP_VRN

如果 **node_type** 未知，则 AP_LEARN_NODE X 须指定。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

frsn w ？ u 减 J 3 序号。如果为非 c 值，那 4 v P 带大 Z 或 H Z C 值 D FRSN D 拓扑 a 构规拱巩构构勾苟 拱 乖鼓龚骨供乖龟汞蛊龚狗构苟规拱巩构构勾苟 拱 乖鼓龚骨

/ F 1 3 1 T f 1 0 T D (c) / F 4 3 1 T f 1 . 9 8 9 3 0 2 8 1 T f 0 3 9 1 T f 1 0 T D (n 1 T

SFS_NN_TOPOLOGY_NODE

nn_topology_node_detail.frsn
w ? u 减 J 3 序号。 b 指 > K 在 > X Z c 资源 | 新 D 最后 1 间。

nn_topology_node_detail.rsn
资源 3 序号。 G I 5 P C 资源 D 网 g Z c 分配 D。

nn_topology_node_detail.rar
网 g Z c 7 I D = 加 V 9 性。

nn_topology_node_detail.status
C 字段指定 K Z c D 状，" R 能够 G AP_UNCONGESTED 或下 P 一 v 或多 v 一起或运 c D 值:

AP_CONGESTED
大 Z 在 START_NODE 动词中指定 D **isr_sessions_upper_threshold** ISR 会话 D 号 k 。

AP_IRR_DEPLETED
在 START_NODE 动词 D N } **max_isr_sessions** 中指定 D，已达 = 最大值 ISR 会话 D 号 k 。

AP_ERR_DEPLETED
已达 = 最大指定值 D 端 c 会话 D 号 k 。

AP_QUIESCING
发 v AP_QUIENCE 或 AP_QUIENCE_ISR 类型 D STOP_NODE 。

nn_topology_node_detail.function_support
C 字段指定 K 支 V 哪一种功能。 C 字段 I 能 G 下 P 一 v 或多 v :

AP_BORDER_NODE
支 V _ g Z c 功能。

AP_CDS
支 V 中心目 < 服务器。

AP_GATEWAY
C Z c G 网关 Z c （ C 功能 P 未 a 构性定义)。

AP_ISR
C Z c 支 V = i 会话 7 I 选择。

AP_HPR
C Z c 支 V = i 会话 7 I 选择。

AP_RTP_TOWER
C Z c 支 V HPR D RTP ~。

AP_CONTROL_OVER_RTP_TOWER
C Z c 支 V RTP ~ O D X 制 w。

"： AP_CONTROL_OVER_RTP_TOWER Z c 同 AP_HPR 和 AP_RTP_TOWER = v Z c D h 置相关。

如果动词未 I 功执行，则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LIST_OPTION

AP_INVALID_NODE
AP_INVALID_LIST_OPTIONS

提供N数

当 restore 等Z AP_YES 时提供N数

&C L序a供下P N }:

opcode

AP_SFS_NN_TOPOLOGY_NODE

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C&C L序能够m加}] = VCB D末端。

num_entries

返回项D最大} ?。项D } ? + ；会，过b v值。 c值m >无限制。

restore

m w G否&C恢4（AP_YES）或存储（AP_NO）信息D j 志位。在b种i v下，置为 AP_NO 。

nn_topology_node_detail.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移?。

nn_topology_node_detail.node_name

来自网g拓扑a构}] b D网g完{ 7 6 Z c {。 b v {字G I EBCDIC U q R n d D 17 字Z D相Z X制c {。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。(? v { F D $度最多I为8 v字Z， R；含6入U q。）C字段v对= APPN Z c D 4 7 G J 1 D " R否则+忽T。如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽T C字段。

nn_topology_node_detail.node_type

Z c D类型。 | G下P之一:

AP_NETWORK_NODE
AP_VRN

nn_topology_node_detail.days_left

从拓扑a构}] b > } C Z c项之0 D l }。如果C Z c ；G > X Z c，则C字段X须置为大Z c D值。

nn_topology_node_detail.frsn

w ? u减J 3序号。 b指> K在> X Z c资源 | 新D最后1间。

nn_topology_node_detail.rsn

资源3序号。 G I 5 P C资源D网g Z c分配D。

nn_topology_node_detail.rar

网g Z c 7 I D =加V 9性。

SFS_NN_TOPOLOGY_NODE

nn_topology_node_detail.status

C字段指定K Z c D状，" R能够G AP_UNCONGESTED 或下P一v或多v一起或运c D值:

AP_CONGESTED

大Z在 START_NODE 动词中指定D **isr_sessions_upper_threshold** ISR 会话D号k。

AP_IRR_DEPLETED

在 START_NODE 动词D N } **max_isr_sessions** 中指定D，已达=最大值 ISR 会话D号k。

AP_ERR_DEPLETED

已达=最大指定值D端c会话D号k。

AP_QUIESCING

发v AP_QUIENCE 或 AP_QUIENCE_ISR 类型D STOP_NODE 。

nn_topology_node_detail.function_support

C字段指定K支V哪一种功能。 C字段I能G下P一v或多v:

AP_BORDER_NODE

支V _ g Z c功能。

AP_CDS

支V中心目<服务器。

AP_GATEWAY

C Z c G网关Z c （C功能P未a构性定义)。

AP_ISR

C Z c支V = i会话7 I选择。

AP_HPR

C Z c支V = i会话7 I选择。

AP_RTP_TOWER

C Z c支V HPR D RTP ~。

AP_CONTROL_OVER_RTP_TOWER

C Z c支V RTP ~ O D X制w。

": AP_CONTROL_OVER_RTP_TOWER Z c同 AP_HPR 和 AP_RTP_TOWER = v Z c D h置相关。

node_type

Z c D类型。 b v Z c置为下P之一:

AP_NETWORK_NODE
AP_VRN

如果 **node_type** 未知，则 AP_LEARN_NODE X须指定。如果 **list_options** 置为 AP_FIRST_IN_LIST ，则忽T C字段。

frsn w ? u减J 3序号。如果为非c值，那4 v P带大Z或H Z C值D FRSN D拓扑a构资源返回。

返回N数

如果C动词I功执行，则L序返回下PN}:

primary_rc
AP_OK

secondary_rc
AP_INVALID_DAYS_LEFT

如果C动词因N}错误而未能执行，则L序返回下PN}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DAYS_LEFT

如果C动词因N}错误而未能执行，则L序返回下PN}:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DAYS_LEFT

如果C动词因k hte相关D START_NODE N}；Ph置，则L序返回下PN}:

primary_rc
AP_FUNCTION_NOT_SUPPORTED

secondary_rc
AP_INVALID_DAYS_LEFT

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_INVALID_VERB

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

SFS_NN_TOPOLOGY_TG

": C动词已 I SAFE_STORE_TOPOLOGY f代, " R v为K k先0 f > D L序D兼容性而# t。

?一v网g Z c维护着V P关Z y P网g Z c、VRN 和网g Z c =网g中D网g Z c TG 信息D网g Z c拓扑a构}] b。如果Z c重新启动, SFS_NN_TOPOLOGY_NODE 动词C Z 2全X储存能够以后访问D拓扑a构}] b Z c d入。**restore** j志位C Z指>信息G否存储（AP_NO）或访问（AP_YES）。C动词9 C **topology_tg_detail** 2 G。

为K # t P关X定网g Z c D信息或为K # t } v 『chunks』 中D信息, **owner**、**owner_type**、**dest**、**dest_type** 和 **tg_num** 字段&C # t。

否则（如果 **list_options** 字段置为 AP_FIRST_IN_LIST）, +忽T C字段。k N阅 Z 12页D『i询Z c』, 以获取关Z如何9 C P m q = D 3 0知6。

C P m 4 U**owner**、**owner_type**、**dest**、**dest_type**、**tg_num** 和 **frsn** 。**owner_type** 和 **dest** {字W先4 {字$度排序, " R然后对Z相同$度D {字4 ASCII k词d 3序定序（y] IBM D 6611 APPN MIB 定序）。**owner_type** 和 **dest** D次序G: AP_NETWORK_NODE, 然后 AP_VRN 。**tg_num** 和 **frsn** 4 }字排序。

- 如果选择K AP_LIST_INCLUSIVE , 则返回P m从b v {字D Z一v P效记< * <。
- 如果选择K AP_LIST_FROM_NEXT , 则C P m +从I C指定{ D Z一v P效记< * <。

注意如果 **frsn** 字段置为非c值, 则v P _ Z C值D带P w ? u减3序号（FRSN）D }] b项返回。b样允许在 『chunks』 D一v号k通过W次C = C Z c D 1 0 FRSN 1, 返回一致D拓扑a构}] b。工作如下:

1. 发v返回Z c 1 0 FRSN D QUERY_NODE 。
2. 为C = 『chunks』 中D y P }] b项, ! I 能D发v许多SFS_NN_TOPOLOGY_NODE（FRSN 置为c）。
3. 再次发v QUERY_NODE " R H O新D FRSN k在Z一W段返回D FRSN 。
4. 如果= v FRSN ;同, 那4在}] b中y D d D FRSN, 发v带P置为大Z Z一W段a供D FRSN D FRSN D SFS_NN_TOPOLOGY_NODE 。

VCB a构

```
typedef struct sfs_nn_topology_tg
{
        unsigned short  opcode;             /* verb operation code           */
        unsigned char   reserv2;            /* reserved                      */
        unsigned char   format;             /* format                        */
        unsigned short  primary_rc;         /* primary return code           */
        unsigned long   secondary_rc;       /* secondary return code         */
        unsigned char   *buf_ptr;           /* pointer to buffer             */
        unsigned long   buf_size;           /* buffer size                   */
        unsigned long   total_buf_size;     /* total buffer size required    */
        unsigned short  num_entries;        /* number of entries             */
        unsigned short  total_num_entries;  /* total number of entries       */
        unsigned char   list_options;       /* listing options               */
        unsigned char   restore;            /* store or restore;             */
        unsigned char   owner[17];          /* network qualified             */
                                            /* node name                     */
        unsigned char   owner_type;         /* node type                     */
        unsigned char   dest[17];           /* TG destination node           */
```

```
        unsigned char   dest_type;          /* TG destination node type      */
        unsigned char   tg_num;             /* TG number                     */
        unsigned char   reserv1;            /* reserved                      */
        unsigned long   frsn;               /* flow-reduction sequence       */
                                            /* number                        */
} SFS_NN_TOPOLOGY_TG;

typedef struct nn_topology_tg_detail
{
        unsigned short  overlay_size;       /* size of this entry            */
        unsigned char   owner[17];          /* network qualified             */
        unsigned char   owner_type;         /* node type                     */
        unsigned char   dest[17];           /* TG destination node           */
        unsigned char   dest_type           /* TG destination node type      */
        unsigned char   tg_num;             /* TG number                     */
        unsigned char   reserv3[1];         /* reserved                      */
        unsigned long   frsn;               /* flow reduction sequence number*/
        unsigned short  days_left           /* days left in database         */
        LINK_ADDRESS    dlc_data;           /* DLC signalling data           */
        unsigned long   rsn;                /* resource sequence number      */
        unsigned char   status;             /* node status                   */
        TG_DEFINED_CHAR tg_chars;           /* TG characteristics            */
        unsigned char   reserva[20];        /* reserved                     */
}TOPOLOGY_TG_DETAIL;

typedef struct link_address
{
        unsigned short  length              /* length                        */
        unsigned short  reserve1;           /* reserved                      */
        unsigned char   address[MAX_LINK_ADDR_LEN];
                                            /* address                       */
}LINK_ADDRESS;
```

提供N数

当 **restore** 等Z **AP_NO** 时提供N数

&C L序a供下P N }:

opcode

AP_SFS_NN_TOPOLOGY_TG

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

buf_ptr

指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下，C&C L序能够m加}] = VCB D末端。

buf_size

a供D缓e大小。返回}] + ；会，过b v大小。

num_entries

返回项D最大} ?。项D } ? + ；会，过b v值。 c值m > 无限制。

list_options

m w在P m信息中& C返回2 4 。

owner、**owner_type**、**dest**、**dest_type**、**tg_num** 和 **frsn** 指定K（k N阅下P N }）m > C Z指定返回D 5际信息启动c D w引值。

AP_FIRST_IN_LIST

忽T w引值" R返回P m从P m中D Z一项* <。

AP_LIST_FROM_NEXT
在那项 I a 供 D w 引值指定之后，返回 P m 从 P m 中 D 下一项 * <。

AP_LIST_INCLUSIVE
返回 P m 从 I w 引值指定 D 一项 * <。

restore
m w G 否 & C 恢 4（AP_YES）或存储（AP_NO）信息 D j 志位。在 b 种 i v 下，置为 AP_NO 。

owner TG D 发起 Z c { 字（总置为 > X Z c { ）。b v { 字 G I EBCDIC U q R n d D 17 字 Z D 相 Z X 制 c { 。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q。）C 字段 v 对 = APPN Z c D 4 7 G J 1 D " R 否则 + 忽 T。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

owner_type
Z c D 类型。b v Z c 置为下 P 之一:

AP_NETWORK_NODE
AP_VRN

如果 **owner_type** 未知，则 AP_LEARN_NODE X 须指定。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

dest TG D 完 { 7 6 目 D Z c { 。b v { 字 G I EBCDIC U q R n d D 17 字 Z D 相 Z X 制 c { 。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q。）C 字段 v 对 = APPN Z c D 4 7 G J 1 D " R 否则 + 忽 T。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

dest_type
Z c D 类型。b v Z c 置为下 P 之一:

AP_NETWORK_NODE
AP_VRN

如果 **dest_type** 未知，则 AP_LEARN_NODE X 须指定。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

tg_num
同 TG 相关 * D } ?。如果 **list_options** 置为 AP_FIRST_IN_LIST ， 则忽 T C 字段。

frsn w ? u 减 J 3 序号。如果为非 c 值，那 4 v P 带大 Z 或 H Z C 值 D FRSN D 拓扑 a 构资源返回。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N }:

primary_rc
AP_OK

buf_size
在缓 e 中返回 D 信息 $ 度。

total_buf_size

返回值 5 w K + 会需要 C Z 返回 y P P v D k s 信息 D 缓 e 大小。 I 以 _ Z **buf_size** 。

num_entries

5 际返回 D 项 D } 目。

total_num_entries

I 能已 - 返回项 D 总 } 。 I 以 _ Z **num_entries** 。

topology_tg_detail.overlay_size

C 项中 D 字 Z } , 即 = 下一 v 返回项（如果 P D 话）D 偏移 ? 。

topology_detail.owner

TG D 发起 Z c { 字。 b v { 字 G I EBCDIC U q R n d D 17 字 Z D 相 Z X 制 c { 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , R ; 含 6 入 U q 。）

topology_tg_detail.owner_type

Z c D 类型。 | G 下 P 之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_detail.dest

TG D 完 { 7 6 目 D Z c { 。 b v { 字 G I EBCDIC U q R n d D 17 字 Z D 相 Z X 制 c { 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , R ; 含 6 入 U q 。）C 字段 v 对 = APPN Z c D 4 7 G J 1 D " R 否则 + 忽 T 。 如果 **list_options** 置为 AP_FIRST_IN_LIST , 则忽 T C 字段。

topology_tg_detail.dest_type

Z c D 类型。 | G 下 P 之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_detail.tg_num

同 TG 相关 * D 号 k 。

topology_tg_detail.frsn

w ? u 减 J 3 序号。 5 w K 在 > X Z c 资源 | 新 D 最后 1 间。

topology_tg_detail.days_left

在 C Z c # t 在拓扑 a 构 }] b 中 D l } > } 之 0 , } 非 | D 存在能够确认。 如果 I **owner** 字段指定 D Z c ; G > X Z c , 则 C 字段 X 须置为大 Z c D 植。

topology_tg_detail.dlc_data.length

X 址 $ 度。

topology_tg_detail.dlc_data.address

X 址。

topology_tg_detail.rsn

资源 3 序号。 G I 5 P C 资源 D 网 g Z c 分配 D 。

topology_tg_detail.status

C 字段指定 K TG D 状 , 。 C 字段 I 能 G 下 P 一 v 或多 v 一起做或运 c D 值:

AP_TG_OPERATIVE
AP_TG_CP_CP_SESSIONS
AP_TG_QUIESCING
AP_TG_HPR
AP_TG_RTP
AP_NONE

topology_tg_detail.tg_chars
TG D X性。 k N阅 Z 35页D『DEFINE_CN』 ， 以获C其 | 信息。

返回N数

如果C动词因N } 错误而未能执行， 则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_TG

AP_INVALID_ORIGIN_NODE
AP_INVALID_LIST_OPTION

如果动词未I功执行， 则L序返回下P N }:

primary_rc
AP_OK

提供N数

当 restore 等Z AP_YES 时提供N数

&C L序a供下P N }:

opcode
AP_SFS_NN_TOPOLOGY_TG

format
j 6 VCB D q =。 + C字段h置为c， 以指定O f P v D VCB D f >。

buf_ptr
指向能够写入P m信息缓e D指k。在 **buf_ptr** X须置为 NULL D i v下， C&C L序能够m加 }] = VCB D末端。

num_entries
返回项D最大 } ?。项D } ? + ；会，过b v值。 c值m > 无限制。

buf_size
在缓e中返回D信息$度。

restore
m w G否&C恢4（AP_YES）或存储（AP_NO）信息D j 志位。在b种 i v下， 置为 AP_NO 。

total_num_entries
I能已 - 返回项D总 }。 I以 _ Z **num_entries** 。

topology_tg_detail.overlay_size

C项中D字Z }，即=下一v返回项（如果P D话）D偏移？。 1 **restore** H Z AP_NO 1， b X须同返回值 **overlay_size** 一样。

topology_detail.owner

TG D发起Z c { 字。 b v { 字G I EBCDIC U q R n d D 17 字Z D相Z X制c { 。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， R；含6入U q。）

topology_tg_detail.owner_type

5 P TG D Z c类型。 | G下P之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_detail.dest

TG D完{ 7 6目D Z c { 。 b v { 字G I EBCDIC U q R n d D 17 字Z D相Z X制c { 。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， R；含6入U q。）C字段v对=APPN Z c D 4 7 G J 1 D " R否则+忽T。如果 **list_options** 置为AP_FIRST_IN_LIST ，则忽T C字段。

topology_tg_detail.dest_type

Z c D类型。 | G下P之一:

AP_NETWORK_NODE
AP_VRN

topology_tg_detail.tg_num

同 TG 相关*D号k。

topology_tg_detail.frsn

w？u减J 3序号。 5 w K在>X Z c资源|新D最后1间。

topology_tg_detail.days_left

在C Z c # t在拓扑a构}] b中D l } > }之0， }非| D存在能够确认。如果I **owner** 字段指定D Z c ；G>X Z c，则C字段X须置为大Z c D植。

topology_tg_detail.dlc_data.length

X址$度。

topology_tg_detail.dlc_data.address

X址。

topology_tg_detail.rsn

资源3序号。 G I 5 P C资源D网g Z c分配D。

topology_tg_detail.status

C字段指定K TG D状，。 C字段I能G下P一v或多v一起做或运c D值:

AP_TG_OPERATIVE
AP_TG_CP_CP_SESSIONS
AP_TG_QUIESCING
AP_TG_HPR
AP_TG_RTP
AP_NONE

SFS_NN_TOPOLOGY_TG

topology_tg_detail.tg_chars
TG D X性。 k N阅 Z 35页D『DEFINE_CN』 ， 以获C其 | 信息。

返回N数

如果C动词 I 功执行， 则L序返回下P N }:

primary_rc
AP_OK

如果C动词因N } 错误而未能执行， 则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_DAYS_LEFT

如果C动词因 } 确D START_NODE N } ； P发M而未能执行， 则L序返回下P N }:

primary_rc
AP_FUNCTION_NOT_SUPPORTED

如果C动词因系统未内 (网 g Z c 支V而未能执行， 则L序返回下P N }:

primary_rc
AP_INVALID_VERB

如果C动词因Z c P未启动而未能执行， 则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因系统错误而未能执行， 则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSYEM_ERROR

第8章 会话限制动词

> B Z h v K C Z u < 化、 D d 或 4 位会话限制D动词。

© Copyright IBM Corp. 1989, 1997, 1998

CHANGE_SESSION_LIMIT

CHANGE_SESSION_LIMIT 动词 k s D d X定模 = D会话限制（或会话组）。作为处理 C动词D a 果，会话能够激活或M放。

VCB a 构

```
typedef struct change_session_limit
{
   unsigned short  opcode;              /* verb operation code       */
   unsigned char   reserv2;             /* reserved                  */
   unsigned char   format;              /* format                    */
   unsigned short  primary_rc;          /* primary return code       */
   unsigned long   secondary_rc;        /* secondary return code     */
   unsigned char   lu_name[8];          /* local LU name             */
   unsigned char   lu_alias[8];         /* local LU alias            */
   unsigned char   plu_alias[8];        /* partner LU alias          */
   unsigned char   fqplu_name[17];      /* fully qualified partner   */
                                        /* LU name                   */
   unsigned char   reserv3;             /* reserved                  */
   unsigned char   mode_name[8];        /* mode name                 */
   unsigned char   reserv3a;            /* reserved                  */
   unsigned char   set_negotiable;      /* set max negotiable limit? */
   unsigned short  plu_mode_session_limit;
                                        /* session limit             */
   unsigned short  min_conwinners_source;
                                        /* min source contention     */
                                        /* winner sessions           */
   unsigned short  min_conwinners_target;
                                        /* min target contention     */
                                        /* winner sessions           */
   unsigned short  auto_act;            /* auto activation limit     */
   unsigned char   responsible;         /* responsible indicator     */
   unsigned char   reserv4[3];          /* reserved                  */
   unsigned long   sense_data;          /* sense data                */
} CHANGE_SESSION_LIMIT;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_CHANGE_SESSION_LIMIT

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

lu_name

> X LU k s C Z D d 会话限制D LU D { 字。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU 。

lu_alias

> X LU k s C Z D d 会话限制D p { 。 b G v 以 I 显 > D > X字符集m > D 8 字Z D字符串。 v 1 **lu_name** 字段 h 置为全 c 1， | E P 效，在 b 种 i v 下。 y P 8 v字Z都G P 效D " R X须 h 置。如果 **lu_name** 和 **lu_alias** = v 字段置为全 c，则C动词转发 x k X制 c 相关 * D LU（缺！ LU）。

plu_alias

为 > X LU y 知D伙 i LU D p { 。 b v { 字X须 k 在配置过L中 (" D伙 i

LU D { 字匹配。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 y P D 8 v 字 Z G P 效 D " R X 须 h 置。如果 C 字段置为全 c，则 **fqplu_name** 字段 + C Z 指定需要 D 伙 i LU 。

fqplu_name

伙 i LU D 完 { 7 6 LU D { 字。 { F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。（ ? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q。）v 1 **plu_alias** 字段 h 置为全 c 1， | E P 效。

mode_name

在配置过 L 中定义 D 一组网 g X 性 D { F。 b G v 8 字 Z D 字母 } 字集 D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d。

SNASVCMG 和 CPSVCMG 模 = 限制；能 D d。**Set_negotiable** 指定 K G 否修 D C 模 = D 最大 I 协 L 会话限制以 d 为 **plu_mode_session_limit** 。

set_negotiable

指定 K G 否修 D C 模 = D 最大 I 协 L 会话限制以 d 为 **plu_mode_session_limit** 。

AP_YES
AP_NO

plu_mode_session_limit

C 模 = k s D 会话限制 D 总 } 。 5 际会话限制（能够 k 伙 i LU 协 L ），即在 C 模 = 下 > X LU 和伙 i LU 之间支 V D 会话许 I D 最大 } ? 。

min_conwinners_source

C 模 = 下伙 i LU G ： y $ 方 D 会话 D 最小 } ? 。

min_conwinners_target

C 模 = 下伙 i LU G ： y $ 方 D 会话 D 最小 } ? 。

auto_act

在会话限制 D d 后自动激活 D 会话 } ? 。自动激活会话 D 5 际 } ? G C 值 D 最小值和 > X LU D ： y $ 方 D 最小协 L } ? 。 1 } # M 放 D 会话（指定 AP_DEACT_NORMAL ）在 C 限制之下 1，激活 D 新会话增加 = C 限制。

responsible

指 > 在会话限制 D d 之后源（ > X ）或目 j （伙 i ）LU G 否对 M 放会话 x 行响 &（AP_SOURCE 或 AP_TARGET）。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

secondary_rc

AP_AS_SPECIFIED

AP_AS_NEGOTIATED

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_LU_MODE_SESSION_LIMIT_ZERO

AP_EXCEEDS_MAX_ALLOWED
AP_INVALID_MODE_NAME
AP_INVALID_PLU_NAME
AP_INVALID_RESPONSIBLE
AP_INVALID_SET_NEGOTIABLE
AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS

如果C动词因状，错误而未能执行，则L序返回下P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_MODE_RESET

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因分配错误而未能执行，则L序返回下P N }:

primary_rc
AP_ALLOCATION_ERROR

secondary_rc
AP_ALLOCATION_FAILURE_NO_RETRY

sense_data
k分配错误相关* D检b }]。

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

如果C动词因错误而未能执行，则L序返回下P N }:

primary_rc
AP_CONV_FAILURE_NO_RETRY

AP_CNOS_PARTNER_LU_REJECT

secondary_rc
AP_CNOS_COMMAND_RACE_REJECT

AP_CNOS_MODE_NAME_REJECT

INITIALIZE_SESSION_LIMIT

INITIALIZE_SESSION_LIMIT 动词 u < 化 K 模 = 会话限制。

VCB a 构

```
typedef struct initialize_session_limit
{
   unsigned short  opcode;              /* verb operation code       */
   unsigned char   reserv2;             /* reserved                  */
   unsigned char   format;              /* format                    */
   unsigned short  primary_rc;          /* primary return code       */
   unsigned long   secondary_rc;        /* secondary return code     */
   unsigned char   lu_name[8];          /* local LU name             */
   unsigned char   lu_alias[8];         /* local LU alias            */
   unsigned char   plu_alias[8];        /* partner                   */
   unsigned char   fqplu_name[17];      /* fully qualified partner   */
                                        /* LU name                   */
   unsigned char   reserv3;             /* reserved                  */
   unsigned char   mode_name[8];        /* mode name                 */
   unsigned char   reserv3a;            /* reserved                  */
   unsigned char   set_negotiable;      /* set max negotiable limit? */
   unsigned short  plu_mode_session_limit;
                                        /* session limit             */
   unsigned short  min_conwinners_source;
                                        /* min source contention     */
                                        /* winner sessions           */
   unsigned short  min_conwinners_target;
                                        /* min target contention     */
                                        /* winner sessions           */
   unsigned short  auto_act;            /* auto activation limit     */
   unsigned char   reserv4[4];          /* reserved                  */
   unsigned long   sense_data;          /* sense data                */
} INITIALIZE_SESSION_LIMIT;
```

提供 N 数

& C L 序 a 供下 P N }:

opcode

AP_INITIALIZE_SESSION_LIMIT

format

j 6 VCB D q =。 + C 字段 h 置为 c，以指定 O f P v D VCB D f >。

lu_name

> X LU k s C Z u < 化会话限制 D LU D { 字。C { F G v 8 字 Z D A 型 EBCDIC 字符串。如果 C 字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU。

lu_alias

> X LU k s C Z u < 化会话限制 D p { 。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。v 1 **lu_name** 字段 h 置为全 c 1， | E P 效，在 b 种 i v 下。y P 8 v 字 Z 都 G P 效 D " R X 须 h 置。如果 **lu_name** 和 **lu_alias** = v 字段置为全 c，则 C 动词转发 x k X 制 c 相关 * D LU（缺！ LU）。

plu_alias

为 > X LU y 知 D 伙 i LU D p { 。 b v { 字 X 须 k 在配置过 L 中 (" D 伙 i

LU D { 字匹配。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 y P D 8 v 字 Z G P 效 D " R X 须 h 置。如果 C 字段置为全 c，则 **fqplu_name** 字段 + C Z 指定需要 D 伙 i LU 。

fqplu_name

伙 i LU D 完 { 7 6 LU D { 字。 { F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。(? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q。) v 1 **plu_alias** 字段 h 置为全 c 1， | E P 效。

mode_name

在配置过 L 中定义 D 一组网 g X 性 D { F。 b G v 8 字 Z D 字母 } 字集 D A 型 EBCDIC 字符串（以一 v 字母 * <)， I EBCDIC U q R n d。

如果在 C 字段中 a 供 { 为 SNASVCMG 或 CPSVCMG D 模 = 之一， " R 限 Z 取 **plu_mode_session_limit 2**、**min_conwinners_source 1** 和 **min_conwinners target 1** 之外 D 值，则 \ x C 动词。

set_negotiable

指定 K G 否修 D C 模 = D 最大 I 协 L 会话限制以 d 为 **plu_mode_session_limit** 。

AP_YES
AP_NO

plu_mode_session_limit

C 模 = k s D 会话限制 D 总 }。 5 际会话限制（能够 k 伙 i LU 协 L)，即在 C 模 = 下 > X LU 和伙 i LU 之间支 V D 会话许 I D 最大 } ?。 b 样 X 须 h 置一 v 在范围 1 = 32 767 间 D 值。

min_conwinners_source

C 模 = 下伙 i LU G : y $ 方 D 会话 D 最小 } ?。 b 样 X 须 h 置一 v 在范围 0 = 32 767 间 D 值。

min_conwinners_target

C 模 = 下伙 i LU G : y $ 方 D 会话 D 最小 } ?。 b 样 X 须 h 置一 v 在范围 0 = 32 767 间 D 值。

auto_act

在会话限制 D d 后自动激活 D 会话 } ?。自动激活会话 D 5 际 } ? G C 值 D 最小值和 > X LU D : y $ 方 D 最小协 L } ?。 1 } # M 放 D 会话（指定 AP_DEACT_NORMAL ）在 C 限制之下 1，激活 D 新会话增加 = C 限制。 b 样 X 须 h 置一 v 在范围 0 = 32 767 间 D 值。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

secondary_rc

AP_AS_SPECIFIED

AP_AS_NEGOTIATED

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_CANT_CHANGE_TO_ZERO

AP_EXCEEDS_MAX_ALLOWED
AP_INVALID_SET_NEGOTIABLE
AP_INVALID_PLU_NAME
AP_INVALID_MODE_NAME
AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS
AP_INVALID_SCVMG_LIMITS

如果C动词因状，错误而未能执行，则L序返回下P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_MODE_NOT_RESET

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因分配错误而未能执行，则L序返回下P N }:

primary_rc
AP_ALLOCATION_ERROR

secondary_rc
AP_ALLOCATION_FAILURE_NO_RETRY

sense_data
k分配错误相关* D检b }]。

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

如果C动词因错误而未能执行，则L序返回下P N }:

primary_rc
AP_CONV_FAILURE_NO_RETRY

AP_CNOS_PARTNER_LU_REJECT

secondary_rc

AP_CNOS_COMMAND_RACE_REJECT

AP_CNOS_MODE_NAME_REJECT

RESET_SESSION_LIMIT

RESET_SESSION_LIMIT 动词 k s 模 = 会话限制 4 位。

VCB a 构

```
typedef struct reset_session_limit
{
   unsigned short  opcode;           /* verb operation code        */
   unsigned char   reserv2;          /* reserved                   */
   unsigned char   format;           /* format                     */
   unsigned short  primary_rc;       /* primary return code        */
   unsigned long   secondary_rc;     /* secondary return code      */
   unsigned char   lu_name[8];       /* local LU name              */
   unsigned char   lu_alias[8];      /* local LU alias             */
   unsigned char   plu_alias[8];     /* partner LU alias           */
   unsigned char   fqplu_name[17];   /* fully qual partner LU name */
   unsigned char   reserv3;          /* reserved                   */
   unsigned char   mode_name[8];     /* mode name                  */
   unsigned char   mode_name_select; /* select mode name           */
   unsigned char   set_negotiable;   /* set max negotiable limit?  */
   unsigned char   reserv4[8];       /* reserved                   */
   unsigned char   responsible;      /* responsible                */
   unsigned char   drain_source;     /* drain source               */
   unsigned char   drain_target;     /* drain target               */
   unsigned char   force;            /* force                      */
   unsigned long   sense_data;       /* sense data                 */
} RESET_SESSION_LIMIT;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_RESET_SESSION_LIMIT

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

lu_name

> X LU k s C Z D d 会话限制D LU D { 字。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU。

lu_alias

> X LU k s C Z 4 位会话限制D p { 。 b G v 以 I 显 > D > X字符集 m > D 8 字Z D字符串。 v 1 **lu_name** 字段 h 置为全 c 1， | E P 效，在 b 种 i v 下。 y P 8 v 字 Z 都G P 效D " R X 须 h 置。如果 b 置为全 c， C 动词 + 转发 x k X制 c 相关 * D LU（缺！ LU）。

plu_alias

为 > X LU y 知D伙 i LU D p { 。 b v { 字X须 k 在配置过L中 (" D伙 i LU D { 字匹配。 b G v 在 I 显 > D > X字符集中D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D " R X 须 h 置。如果C字段置为全 c，则 **fqplu_name** 字段 + C Z 指定需要D伙 i LU 。

fqplu_name

伙 i LU D完 { 7 6 LU D { 字。 { F $ 度为 17 v 字 Z， " 以 EBCDIC U

q R n d。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。（？v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q。）如果 **plu_alias** 字段置为全 c，则 C 字段 v 在此 1 P 效。

mode_name
在配置过 L 中定义 D 一组网 g X 性 D { F。 b G v 8 字 Z D 字母 } 字集 D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d。

mode_name_select
选择在 % 独指定模 = 下，或在 > X LU 和伙 i LU 间 D y P 模 = 下 G 否 4 位会话限制。

AP_ONE
AP_ALL

set_negotiable
指定 K G 否修 D C 模 = D 最大 I 协 L 会话限制。

AP_YES
AP_NO

responsible
5 w 在会话限制 D d 之后源（ > X ）或目 j （伙 i ）LU G 否对 M 放会话 x 行响 &（AP_SOURCE 或 AP_TARGET）。

drain_source
1 会话限制 D d 或 4 位 1，指定在 M 放会话之 0，源 LU G 否统计 H 待会话 k s （AP_NO 或 AP_YES）。

drain_target
1 会话限制 D d 或 4 位 1，指定在 M 放会话之 0，目 j LU G 否统计 H 待会话 k s （AP_NO 或 AP_YES）。

force 指定会话限制 G 否置为 c 即 9 CNOS 协 L ' \ （AP_YES 或 AP_NO）。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N }：

primary_rc
AP_OK

secondary_rc
AP_FORCED

AP_AS_SPECIFIED
AP_AS_NEGOTIATED

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }：

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_EXCEEDS_MAX_ALLOWED

AP_INVALID_PLU_NAME
AP_INVALID_MODE_NAME
AP_INVALID_MODE_NAME_SELECT
AP_INVALID_RESPONSIBLE
AP_INVALID_DRAIN_SOURCE
AP_INVALID_DRAIN_TARGET
AP_INVALID_FORCE
AP_INVALID_SET_NEGOTIABLE
AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS

如果C动词因状，错误而未能执行，则L序返回下PN}:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_MODE_RESET

如果C动词因Z c P未启动而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下PN}:

primary_rc
AP_NODE_STOPPING

如果C动词因分配错误而未能执行，则L序返回下PN}:

primary_rc
AP_ALLOCATION_ERROR

secondary_rc
AP_ALLOCATION_FAILURE_NO_RETRY

sense_data
k分配错误相关*D检b }]。

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

如果C动词因错误而未能执行，则L序返回下PN}:

primary_rc
AP_CONV_FAILURE_NO_RETRY

AP_CNOS_PARTNER_LU_REJECT

secondary_rc
AP_CNOS_COMMAND_RACE_REJECT

AP_CNOS_MODE_NAME_REJECT

第9章 Z点Y w功\ API 指导

Z c Y作功能 API 发v指n通知Z c Y作C Z c D d化。指n 9 C下P一c a构:

```
typedef struct indication_hdr
          {
            unsigned short  opcode;         /* verb operation code      */
            unsigned char   reserv2;        /* reserved                 */
            unsigned char   format;         /* format                   */
            unsigned short  primary_rc;     /* primary return code      */
            unsigned long   secondary_rc;   /* secondary return code    */
            unsigned char   data_lost;      /* previous indication lost */
          } INDICATION_HDR;
```

© Copyright IBM Corp. 1989, 1997, 1998

DLC_INDICATION

1 DLC 从活动D转为；活动D，或从；活动D转为活动D 1， z I C指n。

VCB Structure

```
typedef struct dlc_indication
{
   unsigned short  opcode;              /* verb operation code           */
   unsigned char   attributes;          /* verb attributes               */
   unsigned char   format;              /* format                        */
   unsigned short  primary_rc;          /* primary return code           */
   unsigned long   secondary_rc;        /* secondary return code         */
   unsigned char   data_lost;           /* previous indication lost      */
   unsigned char   deactivated;         /* has session been deactivated? */
   unsigned char   dlc_name[8];         /* link station name             */
   unsigned char   description[RD_LEN]; /* resource description          */
   unsigned char   reserva[20];         /* reserved                      */
} DLC_INDICATION;
```

Parameters

opcode

AP_DLC_INDICATION

attributes

动词D t 性。C字段G位字段。Z一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+ C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

secondary_rc

H Z c。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。1内？组件检b = 一次引起0一u指n丢 ' D故O后h置。如果 **data_lost** j志位置为 AP_YES，那4 f后D }] r能置为U。C & C L序&发v一u QUERY | n C以 | 新已丢 ' D信息。

deactivated

1 DLC d为；活动D 1置为 AP_YES。1 DLC d为活动D 1置为 AP_YES。

dlc_name

DLC D { 字。b G v以I显 > D > X字符集m > D 8 字Z D字符串。y P D 8 v字Z G P效D。

description

资源5 w（如在 DEFINE_DLC 中5 w D）。b G v以I显 > D > X字符集m > D 16 字Z D字符串。y P D 16 v字Z G P效D。

DLUR_LU_INDICATION

无 [何 1 DLUR LU 激活或M放，都会 z I C指 n 。 b 样允许一 v 注 a 过D L 序 # t 一份 1 0 活动D DLUR LU P m。

VCB Structure

```
typedef struct dlur_lu_indication
{
   unsigned short  opcode;             /* verb operation code             */
   unsigned char   reserv2;            /* reserved                        */
   unsigned char   format;             /* format                          */
   unsigned short  primary_rc;         /* primary return code             */
   unsigned long   secondary_rc;       /* secondary return code           */
   unsigned char   data_lost;          /* previous indication lost        */
   unsigned char   reason;             /* reason for this indication      */
   unsigned char   lu_name[8];         /* LU name                         */
   unsigned char   pu_name[8];         /* PU name                         */
   unsigned char   nau_address;        /* NAU address                     */
   unsigned char   reserv5[7];         /* reserved                        */
} DLUR_LU_INDICATION;
```

Parameters

opcode

AP_DLUR_LU_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c ，以指定O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为U。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

reason

如果 DLUR LU ! I I DLUS 启动，则置为 AP_ADDED。如果 DLUR LU 要 4 w显X通过 DLUS M放，要 4 ; w显X通过一次 4 7 故O或 PU DM放而M放，则置为 AP_REMOVED。

lu_name

LU D { 字。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ），I EBCDIC U q R n d 。

pu_name

LU 9 C D PU D { 字。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ），I EBCDIC U q R n d 。

nau_address

范围&在 1-255 之间D LU D网 g I 访问%元X址。

DLUR_PU_INDICATION

无 [何 1 DLUR PU 启动或M放，都 z I C指 n 。 b 样允许一 v 注 a 过D L 序 # t 一份 1 0 活动D DLUR PU P m。

VCB Structure

```
typedef struct dlur_pu_indication
{
   unsigned short  opcode;              /* verb operation code            */
   unsigned char   reserv2;             /* reserved                       */
   unsigned char   format;              /* format                         */
   unsigned short  primary_rc;          /* primary return code            */
   unsigned long   secondary_rc;        /* secondary return code          */
   unsigned char   data_lost;           /* previous indication lost       */
   unsigned char   reason;              /* reason for this indication     */
   unsigned char   pu_name[8];          /* PU name                        */
   unsigned char   pu_id[4];            /* PU identifier                  */
   unsigned char   pu_location;         /* downstream or local PU         */
   unsigned char   pu_status;           /* status of the PU               */
   unsigned char   dlus_name[17];       /* current DLUS name              */
   unsigned char   dlus_session_status; /* status of the DLUS pipe        */
   unsigned char   reserv5[2];          /* reserved                       */
} DLUR_PU_INDICATION;
```

Parameters

opcode

AP_DLUR_PU_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c ，以指定O f P v D VCB D f > 。

primary_rc

AP_OK

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为U。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

reason

C指 n D原因。 | G下 P 之一:

AP_ACTIVATION_STARTED

PU } 在启动。

AP_ACTIVATING

PU d 为活动D。

AP_DEACTIVATING

PU d 为 ; 活动D。

AP_FAILED

PU ' \ K。

AP_ACTIVATION_FAILED

PU 无法启动。

pu_name

PU D { 字。 b G v 8 字 Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d。

pu_id 在 DEFINE_INTERNAL_PU | n中定义D或从下N PU D XID 获C D PU j 6符。 b G v 4 字 Z D . y x 制字符串。位 0-11 置为 i 号和位 12 至 31 置为唯一 6 p PU D ID 号。

plu_location

PU D存储%元。 I 能G下 P 之一:

AP_INTERNAL
AP_DOWNSTREAM

dlur_pu_detail.pu_status

PU D状，（如通过 DLUR y 见）。能置为下 P 之一:

AP_RESET_NO_RETRY
PU 处 Z 4 位状，" R + ；会重发。

AP_RESET_RETRY
PU 处 Z 4 位状，" R会重发。

AP_PEND_ACTPU
PU } 在H待从主机来D一 v ACTPU。

AP_PEND_ACTPU_RSP
在转发 ACTPU x PU 之后，DLUR } 在H待 PU 响&。

AP_ACTIVE
PU G活动D。

AP_PEND_DACTPU_RSP
在转发 DACTPU x PU 后，DLUR } 在H待 PU 响&。

AP_PEND_INOP
为K在M放 PU 之 0 a x，DLUR } 在H待 y P X需D B件。

pu_id PU } 在 9 C（或T图 9 C）D DLUS Z c D { 字。 b G v C EBCDIC c 串 * D = v A 型 EBCDIC 字符串组 I D 17 字 Z D字符串，那字符串 G I EBCDIC U q R n d D。（ ? v { F D $ 度最多 I 为 8 v字 Z， R ；含 6 入U q 。）如果 PU 启动 ' \ ， C r + 置为全 c 。

dlur_pu_detail.dlus_session_status

1 0 PU 9 C D DLUS D管@状，。 I 能G下 P 之一:

AP_PENDING_ACTIVE
AP_ACTIVE
AP_PENDING_INACTIVE
AP_INACTIVE

DLUS_INDICATION

1通往 DLUS Z c D管@从活动D转为；活动D（或反之亦然）1， z I C指n。管@D统计信息在管@ d 为；活动D 1 a 供。

VCB Structure

```
typedef struct dlus_indication
{
   unsigned short  opcode;              /* verb operation code             */
   unsigned char   reserv2;             /* reserved                        */
   unsigned char   format;              /* format                          */
   unsigned short  primary_rc;          /* primary return code             */
   unsigned long   secondary_rc;        /* secondary return code           */
   unsigned char   data_lost;           /* previous indication lost        */
   unsigned char   deactivated;         /* has session been deactivated?   */
   unsigned char   dlus_name[17];       /* DLUS name                       */
   unsigned char   reserv1;             /* reserved                        */
   PIPE_STATS      pipe_stats;          /* pipe statistics                 */
   unsigned char   reserva[20];         /* reserved                        */
} DLUS_INDICATION;

typedef struct pipe_stats
{
   unsigned long   reqactpu_sent;       /* REQACTPUs sent to DLUS          */
   unsigned long   reqactpu_rsp_received;
                                        /* RSP(REQACTPU)s received         */
                                        /* from DLUS                       */
   unsigned long   actpu_received;      /* ACTPUs received from DLUS       */
   unsigned long   actpu_rsp_sent;      /* RSP(ACTPU)s sent to DLUS        */
   unsigned long   reqdactpu_sent;      /* REQDACTPUs sent to DLUS         */
   unsigned long   reqdactpu_rsp_received;
                                        /* RSP(REQDACTPU)s received        */
                                        /* from DLUS                       */
   unsigned long   dactpu_received;     /* DACTPUs received from DLUS      */
   unsigned long   dactpu_rsp_sent;     /* RSP(DACTPU)s sent to DLUS       */
   unsigned long   actlu_received;      /* ACTLUs received from DLUS       */
   unsigned long   actlu_rsp_sent;      /* RSP(ACTLU)s sent to DLUS        */
   unsigned long   dactlu_received;     /* DACTLUs received from DLUS      */
   unsigned long   dactlu_rsp_sent;     /* RSP(DACTLU)s sent to DLUS       */
   unsigned long   sscp_pu_mus_rcvd;    /* MUs for SSCP-PU sess received   */
   unsigned long   sscp_pu_mus_sent;    /* MUs for SSCP-PU sessions sent   */
   unsigned long   sscp_lu_mus_rcvd;    /* MUs for SSCP-LU sess received   */
   unsigned long   sscp_lu_mus_sent;    /* MUs for SSCP-LU sessions sent   */
} PIPE_STATS;
```

Parameters

opcode

AP_DLUS_INDICATION

format

j 6 VCB D q =。+ C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

secondary_rc

H Z c。

data_lost

5 w }] G否已 - 丢 '（AP_YES 或 AP_NO）。1内？组件检b = 一次引起0

一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为U。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

deactivated
1 管@ d 为；活动D 1 置为 AP_YES。 1 管@ d 为活动D 1 置为 AP_YES。

dlus_name
DLUS D { 字。 b G v C EBCDIC c 串 * D = v A 型 EBCDIC 字符串组 I D， I EBCDIC U q R n d D 17 字 Z D字符串。（ ? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 人U q。）

pipe_stats.reqactpu_sent
通过管@发M x DLUS D REQACTPU D } ?。

pipe_stats.reqactpu_rsp_received
通过管@从 DLUS S U D RSP（REQACTPU）D } ?。

pipe_stats.actpu_received
通过管@从 DLUS S U D ACTPU D } ?。

pipe_stats.actpu_rsp_sent
通过管@发M x DLUS D RSP（ACTPU）D } ?。

pipe_stats.reqdactpu_sent
通过管@发M x DLUS D REQDACTPU D } ?。

pipe_stats.reqdactpu_rsp_received
通过管@从 DLUS S U D RSP（REQDACTPU）D } ?。

pipe_stats.dactpu_received
通过管@从 DLUS S U D DACTPU D } ?。

pipe_stats.dactpu_rsp_sent
通过管@发M x DLUS D RSP（DACTPU）D } ?。

pipe_stats.actlu_received
通过管@从 DLUS S U D ACTLU D } ?。

pipe_stats.actlu_rsp_sent
通过管@发M x DLUS D RSP（ACTLU）D } ?。

pipe_stats.dactlu_received
通过管@从 DLUS S U D DACTLU D } ?。

pipe_stats.dactlu_rsp_sent
通过管@发M x DLUS D RSP（DACTLU）D } ?。

pipe_stats.sscp_pu_mus_rcvd
通过管@从 DLUS S U D SSCP-PU MU D } ?。

pipe_stats.sscp_pu_mus_sent
通过管@发M x DLUS D SSCP-PU MU D } ?。

pipe_stats.sscp_lu_mus_rcvd
通过管@从 DLUS S U D SSCP-LU MU D } ?。

pipe_stats.sscp_lu_mus_sent
通过管@发 x DLUS D SSCP-LU MU D } ?。

DOWNSTREAM_LU_INDICATION

C动词ｖＪＣＺ通信服务器。

１在下Ｎ LU 和主机之间Ｄ LU-SSCP 会话从；活动Ｄ转为活动Ｄ（或反之亦然）１，或１ PLU-SLU 会话从；活动Ｄ转为活动Ｄ（或反之亦然）１ｚＩＣ指ｎ。LU-SSCP Ｄ统计信息在 LU-SSCP 会话Ｍ放１ａ供 和 PLU-SLU Ｄ统计信息在 PLU-SLU 会话Ｍ放１ａ供。

VCB Structure

```
typedef struct downstream_lu_indication
{
   unsigned short  opcode;               /* verb operation code           */
   unsigned char   attributes;           /* attributes                    */
   unsigned char   format;               /* format                        */
   unsigned short  primary_rc;           /* primary return code           */
   unsigned long   secondary_rc;         /* secondary return code         */
   unsigned char   data_lost;            /* previous indication lost      */
   unsigned char   dspu_name[8];         /* PU Name                       */
   unsigned char   ls_name[8];           /* Link station name             */
   unsigned char   dslu_name[8];         /* LU Name                       */
   unsigned char   description[RD_LEN];  /* resource description          */
   unsigned char   nau_address;          /* NAU address                   */
   unsigned char   lu_sscp_sess_active;
                                         /* Is SSCP session active?       */
   unsigned char   plu_sess_active;      /* Is PLU-SLU session active?    */
   unsigned char   dspu_services;        /* DSPU services                 */
   unsigned char   reserv1;              /* reserved                      */
   SESSION_STATS   lu_sscp_stats;        /* LU-SSCP session statistics    */
   SESSION_STATS   ds_plu_stats;         /* Downstream PLU-SLU sess stats */
   SESSION_STATS   us_plu_stats;         /* Upstream PLU-SLU sess stats   */
} DOWNSTREAM_LU_INDICATION;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;          /* session receive RU size       */
   unsigned short  send_ru_size;         /* session send RU size          */
   unsigned short  max_send_btu_size;    /* max send BTU size             */
   unsigned short  max_rcv_btu_size;     /* max rcv BTU size              */
   unsigned short  max_send_pac_win;     /* max send pacing window size   */
   unsigned short  cur_send_pac_win;     /* curr send pacing window size  */
   unsigned short  max_rcv_pac_win;      /* max rcv pacing window size    */
   unsigned short  cur_rcv_pac_win;      /* curr receive pacing win size  */
   unsigned long   send_data_frames;     /* number of data frames sent    */
   unsigned long   send_fmd_data_frames;
                                         /* num FMD data frames sent      */
   unsigned long   send_data_bytes;      /* number of data bytes sent     */
   unsigned long   rcv_data_frames;      /* num of data frames received   */
   unsigned long   rcv_fmd_data_frames;
                                         /* num FMD data frames received  */
   unsigned long   rcv_data_bytes;       /* num data bytes received       */
   unsigned char   sidh;                 /* session ID high byte          */
   unsigned char   sidl;                 /* session ID low byte           */
   unsigned char   odai;                 /* ODAI bit set                  */
   unsigned char   ls_name[8];           /* Link station name             */
   unsigned char   pacing_type;          /* type of pacing in use         */
} SESSION_STATS;
```

Parameters

opcode

AP_DOWNSTREAM_LU_INDICATION

attributes

动词D t 性。 C字段G位字段。 Z一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

secondary_rc

H Z c。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1内？组件检b = 一次引起0一u指n丢 ' D故O后h置。如果 **data_lost** j志位置为 AP_YES，那4 f后D }] r能置为U。 C & C L序&发v一u QUERY | n C以 | 新已丢 ' D信息。

dspu_name

同下N LU 相关 * D下N PU D { 字。 b G v 8 字Z D字母 } 字集D A 型EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。

ls_name

4 7 > D { 字。 b G v以I显 > D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z都G P效D， R X须h置。

dslu_name

下N LU D { 字。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。

description

资源5 w（如在 DEFINE_DOWNSTREAM_LU 中5 w D ）。

nau_address

范围X需在 1-255 之间D LU D网g I 访问%元X址。

lu_sscp_sess_active

j 6 LU-SSCP 会话 = 下N LU G否G活动D。要4置为 AP_YES，要4置为AP_NO。

plu_sess_active

j 6 PLU-SLU 会话 = 下N LU G否G活动D。要4置为 AP_YES，要4置为AP_NO。

dspu_services

指定 > X Z c a供x下N LU g越4 7 D服务。 C N } 置为下P之一:

AP_PU_CONCENTRATION

> X Z c a供K对下N PU D PU 集a D支V。

DOWNSTREAM_LU__INDICATION

AP_DLUR
> X Z c a 供K对下N PU D DLUR 支V。

lu_sscp_stats.rcv_ru_size
<终# t C字段。

lu_sscp_stats.send_ru_size
<终# t C字段。

lu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

lu_sscp_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

lu_sscp_stats.max_send_pac_win
<终+ C字段 h 置为 c 。

lu_sscp_stats.cur_send_pac_win
<终+ C字段 h 置为 c 。

lu_sscp_stats.max_rcv_pac_win
<终+ C字段 h 置为 c 。

lu_sscp_stats.cur_rcv_pac_win
<终+ C字段 h 置为 c 。

lu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ?。

lu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

lu_sscp_stats.send_data_bytes
发M D } # }] w D字Z }。

lu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

lu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

lu_sscp_stats.rcv_data_bytes
S U D } # }] w D字Z }。

lu_sscp_stats.sidh
Session ID _字Z。

lu_sscp_stats.sidl
Session ID M字Z。

lu_sscp_stats.odai
原<目D X址指>符。 1 (" 会话 1 , 如果> X Z c | 含主 4 7 >, BIND D发M方置C r 为 0; " R如果 BIND D发M方G | 含从 4 7 > D Z c , 则置C r 为 1。

lu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F。 b G v 以 I 显> D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。

lu_sscp_stats.pacing_type
在O N LU-SSCP 会话中 } 在 9 C D S U同 = 类型。 C N } + 取 AP_NONE b v 值。

ds_plu_stats.rcv_ru_size
最大 S U RU 大小。

ds_plu_stats.send_ru_size
最大发M RU 大小。

ds_plu_stats.max_send_btu_size
能发M D最大 BTU 大小。

ds_plu_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

ds_plu_stats.max_send_pac_win
会话中发M w = 窗 Z D最大大小。

ds_plu_stats.cur_send_pac_win
会话中发M w = 窗 Z D 1 0 大小。

ds_plu_stats.max_rcv_pac_win
会话中 S U w = 窗 Z D最大 _ 寸。

ds_plu_stats.cur_rcv_pac_win
会话中 S U w = 窗 Z D 1 0 大小。

ds_plu_stats.send_data_frames
发M D } # w }] 帧 D } ?。

ds_plu_stats.send_fmd_data_frames
发M D } # w FMD }] 帧 D } ?。

ds_plu_stats.send_data_bytes
发M D } # }] w D字 Z }。

ds_plu_stats.rcv_data_frames
S U D } # w }] 帧 D } ?。

ds_plu_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧 D } ?。

ds_plu_stats.rcv_data_bytes
S U D } # }] w D字 Z }。

ds_plu_stats.sidh
Session ID _ 字 Z。

ds_plu_stats.sidl
Session ID M字 Z。

ds_plu_stats.odai
原 < 目 D X址指 > 符。 1 (" 会话 1， 如果 > X Z c | 含主 4 7 >， BIND D发 M方置C r 为 0; " R如果 BIND D发M方G | 含从 4 7 > D Z c， 则置C r 为 1。

ds_plu_stats.ls_name
同统计信息相关 * D 4 7 > { F。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D。

DOWNSTREAM_LU__INDICATION

ds_plu_sscp_stats.pacing_type
在下N PLU-SLU 会话中9 C D S U同=类型。C N }能置为 AP_NONE 或 AP_PACING_FIXED。

us_plu_stats.rcv_ru_size
最大S U RU 大小。

us_plu_stats.send_ru_size
最大发M RU 大小。

us_plu_stats.max_send_btu_size
能发M D最大 BTU 大小。

us_plu_stats.max_rcv_btu_size
能S U D最大 BTU 大小。

us_plu_stats.max_send_pac_win
会话中发M w =窗Z D最大_寸。

us_plu_stats.cur_send_pac_win
会话中发M w =窗Z D 1 0大小。

us_plu_stats.max_rcv_pac_win
会话中S U w =窗Z D最大_寸。

us_plu_stats.cur_rcv_pac_win
会话中S U w =窗Z D 1 0大小。

us_plu_stats.send_data_frames
发M D } # w }]帧D } ?。

us_plu_stats.send_fmd_data_frames
发M D } # w FMD }]帧D } ?。

us_plu_stats.send_data_bytes
发M D } # }] w D字Z }。

us_plu_stats.rcv_data_frames
S U D } # w }]帧D } ?。

us_plu_stats.rcv_fmd_data_frames
S U D } # w FMD }]帧D } ?。

us_plu_stats.rcv_data_bytes
S U D } # }] w D字Z }。

us_plu_stats.sidh
Session ID _字Z。如 **dspu_services** 置为 AP_PU_CONCENTRATION, 则 # t C r。

us_plu_stats.sidl
Session ID M字Z。如 **dspu_services** 置为 AP_PU_CONCENTRATION, 则 # t C r。

us_plu_stats.odai
原<目D X址指>符。1 ("会话1，如果>X Z c |含主4 7 >，BIND D发M方置C r 为 0; "R如果 BIND D发M方G |含从4 7 > D Z c，则置C r 为 1。如 **dspu_services** 置为 AP_PU_CONCENTRATION, 则# t C r。

us_plu_stats.ls_name

同统计信息相关 * D 4 7 > { F。 b G v以 I 显 > D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。如 **dspu_services** 置为 AP_PU_CONCENTRATION, 则# t C r。

us_plu_stats.pacing_type

PLU-SLU 会话中 9 C D S U同 = 类型。 C N } 能取 AP_NONE 或 AP_PACING_FIXED 值。

DOWNSTREAM_PU_INDICATION

C动词ｖＪＣＺ通信服务器。

1在下N PU 和主机之间D PU-SSCP 会话从；活动D转为活动D（或反之亦然）1 z ＩＣ指ｎ。PU-SSCP D统计信息在 PU-SSCP 会话M放１ａ供。

VCB Structure

```
typedef struct downstream_pu_indication
{
   unsigned short  opcode;               /* verb operation code           */
   unsigned char   attributes;           /* attributes                    */
   unsigned char   format;               /* format                        */
   unsigned short  primary_rc;           /* primary return code           */
   unsigned long   secondary_rc;         /* secondary return code         */
   unsigned char   data_lost;            /* previous indication lost      */
   unsigned char   dspu_name[8];         /* PU Name                       */
   unsigned char   description[RD_LEN];  /* resource description          */
   unsigned char   ls_name[8];           /* Link Station name             */
   unsigned char   pu_sscp_sess_active;
                                         /* Is PU-SSCP session active?    */
   unsigned char   dspu_services;        /* DSPU services                 */
   unsigned char   reserv1[2];           /* reserved                      */
   SESSION_STATS   pu_sscp_stats;        /* PU-SSCP session statistics    */
} DOWNSTREAM_PU_INDICATION;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;          /* session receive RU size       */
   unsigned short  send_ru_size;         /* session send RU size          */
   unsigned short  max_send_btu_size;    /* max send BTU size             */
   unsigned short  max_rcv_btu_size;     /* max rcv BTU size              */
   unsigned short  max_send_pac_win;     /* max send pacing window size   */
   unsigned short  cur_send_pac_win;     /* curr send pacing window size  */
   unsigned short  max_rcv_pac_win;      /* max rcv pacing window size    */
   unsigned short  cur_rcv_pac_win;      /* curr receive pacing win size  */
   unsigned long   send_data_frames;     /* number of data frames sent    */
   unsigned long   send_fmd_data_frames;
                                         /* num FMD data frames sent      */
   unsigned long   send_data_bytes;      /* number of data bytes sent     */
   unsigned long   rcv_data_frames;      /* num of data frames received   */
   unsigned long   rcv_fmd_data_frames;
                                         /* num FMD data frames received  */
   unsigned long   rcv_data_bytes;       /* num data bytes received       */
   unsigned char   sidh;                 /* session ID high byte          */
   unsigned char   sidl;                 /* session ID low byte           */
   unsigned char   odai;                 /* ODAI bit set                  */
   unsigned char   ls_name[8];           /* Link station name             */
   unsigned char   pacing;               /* pacing_type                     */
} SESSION_STATS;
```

Parameters

opcode

AP_DOWNSTREAM_PU_INDICATION

attributes

动词Ｄｔ性。C字段G位字段。Ｚ一位｜含要定义资源ＤＩ见性，＂ｋ下Ｐ之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format
j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

primary_rc
AP_OK

secondary_rc
H Z c 。

data_lost
5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1内？组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为U。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

dspu_name
下N PU D { 字。 b G v 8 字 Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d 。

description
资源 5 w（如在 DEFINE_LS 中 5 w D ）。

ls_name
4 7 > D { 字。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D。

pu_sscp_sess_active
指 > PU-SSCP = PU D会话G否激活。要 4 置为 AP_YES，要 4 置为 AP_NO。

dspu_services
指定 > X Z c a 供 x 下N PU g 越 4 7 D服务。 C N } 置为下 P 之一:

AP_PU_CONCENTRATION
> X Z c a 供K对下N PU D PU 集 a D支V。

AP_DLUR
> X Z c a 供K对下N PU D DLUR 支V。

pu_sscp_stats.rcv_ru_size
< 终 # t C字段。

pu_sscp_stats.send_ru_size
< 终 # t C字段。

pu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

pu_sscp_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

pu_sscp_stats.max_send_pac_win
< 终 + C字段 h 置为 c 。

pu_sscp_stats.cur_send_pac_win
< 终 + C字段 h 置为 c 。

DOWNSTREAM_PU__INDICATION

pu_sscp_stats.max_rcv_pac_win
< 终 + C 字段 h 置为 c 。

pu_sscp_stats.cur_rcv_pac_win
< 终 + C 字段 h 置为 c 。

pu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ? 。

pu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ? 。

pu_sscp_stats.send_data_bytes
发M D } # }] w D字 Z } 。

pu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ? 。

pu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ? 。

pu_sscp_stats.rcv_data_bytes
S U D } # }] w D字 Z } 。

pu_sscp_stats.sidh
Session ID _ 字 Z 。

pu_sscp_stats.sidl
Session ID M字 Z 。

pu_sscp_stats.odai
原 < 目D X 址指 > 符。 1 (" 会话 1 , 如果 > X Z c | 含主 4 7 > , BIND D发M方置C r 为 0; " R 如果 BIND D发M方G | 含从 4 7 > D Z c , 则置C r 为 1。

pu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D。

pu_sscp_stats.pacing_type
在O N PU-SSCP 会话中 y 9 C D S U同 = 类型。 C N } + 取 AP_NONE b v 值。

FOCAL_POINT_INDICATION

无 [9 c 获取、D d 或取消，都 z I C 指 n。

VCB Structure

```
typedef struct focal_point_indication
{
   unsigned short  opcode;             /* verb operation code          */
   unsigned char   reserv2;            /* reserved                     */
   unsigned char   format;             /* format                       */
   unsigned short  primary_rc;         /* primary return code          */
   unsigned long   secondary_rc;       /* secondary return code        */
   unsigned char   data_lost;          /* previous indication lost     */
   unsigned char   ms_category[8];     /* Focal point category         */
   unsigned char   fp_fqcp_name[17];   /* Fully qualified focal        */
                                       /* point CP name                */
   unsigned char   ms_appl_name[8];    /* Focal point application name */
   unsigned char   fp_type;            /* type of current focal point  */
   unsigned char   fp_status;          /* status of focal point        */
   unsigned char   fp_routing;         /* type of MDS routing to       */
                                       /* reach FP                     */
   unsigned char   reserva[20];        /* reserved                     */
} FOCAL_POINT_INDICATION;
```

Parameters

opcode

AP_FOCAL_POINT_INDICATION

format

j 6 VCB D q = 。 + C 字段 h 置为 c，以指定 O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G 否已 - 丢 '（AP_YES 或 AP_NO）。1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D 故 O 后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为 U。C & C L 序 & 发 v 一 u QUERY | n C 以 | 新已丢 ' D 信息。

ms_category

9 c 已获取、D d 或取消 D 9 c 类型。对 Z 管理服务 & C L 序，C { 要 4 G 4 v 字 Z a 构 D 定义值（I EBCDIC U q R n d）之一，要 4 G v 8 字 Z D 1134 k D EBCDIC 计 c 定义 {，如同在 *SNA 管理服务* 中 h v D。

fp_fqcp_name

1 0 9 c D 完 { 7 6 X 制 c。{ F $ 度为 17 v 字 Z，" 以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。（? v { F D $ 度最多 I 为 8 v 字 Z，R；含 6 入 U q。）如果 9 c 已取消或 _ ；P f 换（因此；P 1 0 活动 D 9 c），C { 字 + 为全 c 。

ms_appl_name

1 0 9 c D { 字。对 Z 管理服务 & C L 序，C { 要 4 G 4 字 Z 架构 D 定义值

（ I EBCDIC U q R n d ）之一，要 4 G v 8 字 Z D 1134 k D EBCDIC 计 c 定义 {，如同在 *SNA 管理服务* 中 h v D。如果 9 c 已取消或 _ ； P f 换（因此； P 1 0 活动 D 9 c ）， C { 字 + 为全 c 。

fp_type

9 c D 类型。 N < *SNA 管理服务* 以获 C | 详细资 O。

AP_EXPLICIT_PRIMARY_FP
AP_BACKUP_FP
AP_DEFAULT_PRIMARY_FP
AP_DOMAIN_FP
AP_HOST_FP
AP_NO_FP

fp_status

9 c 状，:

AP_NOT_ACTIVE

9 c 已从活动 D d 为；活动 D。

AP_ACTIVE

9 c 已从；活动 D 或暂停活动 D 转为活动 D。

fp_routing

1 9 C MDS 传 M 以发 M }] = 9 c 1 & C L 序 & 指定 D 7 I 类型（只在 9 c 状，为 AP_ACTIVE 1 P 效）:

AP_DEFAULT

缺！ 7 I 选择 C Z 分配 MDS_MU x 9 c 。

AP_DIRECT

MDS_MU + 在一 v 会话 O x 定直 S = 9 c D 7 线。

ISR_INDICATION

C动词v J C Z通信服务器。

1 ISR 会话启动或M放1 z I C指n。 1会话M放1，返回最后D统计信息。 1会话启动1 # t **pri_sess_stats** 和 **sec_sess_stats** r。

VCB Structure

```
typedef struct isr_indication
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   data_lost;         /* previous indication lost     */
   unsigned char   deactivated;       /* has ISR session been         */
                                      /* deactivated?                 */
   FQPCID          fqpcid;            /* fully qualified procedure    */
                                      /* correlator ID                */
   unsigned char   fqplu_name[17];    /* fully qualified primary      */
                                      /* LU name                      */
   unsigned char   fqslu_name[17];    /* fully qualified secondary    */
                                      /* LU name                      */
   unsigned char   mode_name[8];      /* mode name                    */
   unsigned char   cos_name[8];       /* COS name                     */
   unsigned char   transmission_priority;
                                      /* transmission priority        */
   unsigned long   sense_data;        /* sense data                   */
   unsigned char   reserv2a[2];       /* reserved                     */
   SESSION_STATS   pri_sess_stats;    /* primary hop session stats    */
   SESSION_STATS   sec_sess_stats;    /* secondary hop session        */
                                      /* statistics                   */
   unsigned char   reserva[20];       /* reserved                     */
} ISR_INDICATION;

typedef struct fqpcid
{
   unsigned char   pcid[8];           /* pro correlator identifier    */
   unsigned char   fqcp_name[17];     /* orig's network qualified     */
                                      /* CP name                      */
   unsigned char   reserve3[3];       /* reserved                     */
} FQPCID;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;       /* session receive RU size      */
   unsigned short  send_ru_size;      /* session send RU size         */
   unsigned short  max_send_btu_size; /* Maximum send BTU size        */
   unsigned short  max_rcv_btu_size;  /* Maximum rcv BTU size         */
   unsigned short  max_send_pac_win;  /* Max send pacing window size  */
   unsigned short  cur_send_pac_win;  /* Curr send pacing window size */
   unsigned short  max_rcv_pac_win;   /* Max receive pacing win size  */
   unsigned short  cur_rcv_pac_win;   /* Curr rec pacing window size  */
   unsigned long   send_data_frames;  /* Number of data frames sent   */
   unsigned long   send_fmd_data_frames;
                                      /* num of FMD data frames sent  */
   unsigned long   send_data_bytes;   /* Number of data bytes sent    */
   unsigned long   rcv_data_frames;   /* Num data frames received     */
   unsigned long   rcv_fmd_data_frames;
                                      /* num of FMD data frames recvd */
   unsigned long   rcv_data_bytes;    /* Num data bytes received      */
```

```
   unsigned char   sidh;              /* Session ID high byte         */
   unsigned char   sidl;              /* Session ID low byte          */
   unsigned char   odai;              /* ODAI bit set                 */
   unsigned char   ls_name[8];        /* Link station name            */
   unsigned char   pacing_type;       /* type of pacing in use        */
} SESSION_STATS;
```

Parameters

& C L 序 a 供下 P N } :

opcode

AP_ISR_INDICATION

format

j 6 VCB D q = 。 C r 置为 c C 以 5 w 以 O P v D VCB f > 。

primary_rc

AP_OK

data_lost

5 w }] G 否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D 故 O 则置 C r 。如 **data_lost** j 志位置为 AP_YES, 那 4 f 后 D }] r I 置为 U。 C & C L 序 & 发 v 一 u QUERY | n C 以 | 新已丢 ' D 信息。

deactivate

1 ISR 会话 M 放 1 置为 AP_YES。 1 ISR 会话启动 1 置为 AP_NO。

fqpcid.pcid

Procedure Correlator ID。 C ID 为 8 字 Z D . y x 制字符串。

fqpcid.pcid_name

完 { 7 6 X 制 c D { 字。 C { 为 17 字 Z $ R I EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , R ; 含 6 入 U q 。）

fqplu_name

完 { 7 6 LU D { 字（如在 BIND k s 1 ）。 C { 为 17 字 Z $ R I EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , R ; 含 6 入 U q 。）如 **deactivated** 为 AP_YES, 则 C { 全 ? 为 c 。

fqslu_name

完 { 7 6 LU D { 字（如在 BIND k s 1 ）。 C { 为 17 字 Z $ R I EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组 I , 中间以一 v EBCDIC c , S 。（ ? v { F D $ 度最多 I 为 8 v 字 Z , R ; 含 6 入 U q 。）如 **deactivated** 为 AP_YES, 则 C { 全 ? 为 c 。

cos_name

服务类 D { 字。 b G v 8 字 Z D 字母 } 字集 D A 型 EBCDIC 字符串（以一 v 字母 * < ）, I EBCDIC U q R n d 。如 **deactivated** 为 AP_YES, 则 C { 全 ? 为 c 。

transmission_priority

同会话相关 * D 传 d E 先级。如 **deactivated** 为 AP_YES, 则 # t C r 。

sense_data
在b } k s 1发M或S U D检b }]。如 **deactivated** 为 AP_YES，则# t C r。

pri_sess_stats.rcv_ru_size
最大S U RU 大小。

pri_sess_stats.send_ru_size
最大发M RU 大小。

pri_sess_stats.max_send_btu_size
能发M D最大 BTU 大小。

pri_sess_stats.max_rcv_btu_size
能S U D最大 BTU 大小。

pri_sess_stats.max_send_pac_win
会话中发M w = 窗Z D最大 _ 寸。

pri_sess_stats.cur_send_pac_win
会话中发M w = 窗Z D 1 0大小。

pri_sess_stats.max_rcv_pac_win
会话中S U w = 窗Z D最大 _ 寸。

pri_sess_stats.cur_rcv_pac_win
会话中S U w = 窗Z D 1 0大小。

pri_sess_stats.send_data_frames
发M D } # w }] 帧D } ?。

pri_sess_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

pri_sess_stats.send_data_bytes
发M D } # }] w D字Z }。

pri_sess_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

pri_sess_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

pri_sess_stats.rcv_data_bytes
S U D } # }] w D字Z }。

pri_sess_stats.sidh
Session ID _字Z。

pri_sess_stats.sidl
Session ID M字Z。

pri_sess_stats.odai
原<目D X址指>符。 1启动会话1，如果>X Z c |含主4 7 >，则 BIND D发M方置C r为0。如果 BIND D发M方G |含从4 7 >D Z c，则置C r为1。

pri_sess_stats.ls_name
同统计信息相关 * D 4 7 > { F。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 y P D 8 v 字 Z G P 效 D。 C r 能 C Z Q 会话 D 统计信息通过信息 w ? 同 4 7 关 * 起来。

pri_sess_stats.pacing_type
在主会话中 1 0 9 C D S U 同 = 类型。 C N } 能取 AP_NONE、AP_PACING_FIXED 或 AP_PACING_ADAPTIVE H 值。

sec_sess_stats.rcv_ru_size
最大 S U RU 大小。

sec_sess_stats.send_ru_size
最大发 M RU 大小。

sec_sess_stats.max_send_btu_size
能发 M D 最大 BTU 大小。

sec_sess_stats.max_rcv_btu_size
能 S U D 最大 BTU 大小。

sec_sess_stats.max_send_pac_win
会话中发 M w = 窗 Z D 最大 _ 寸。

sec_sess_stats.cur_send_pac_win
会话中发 M w = 窗 Z D 1 0 大小。

sec_sess_stats.max_rcv_pac_win
会话中 S U w = 窗 Z D 最大 _ 寸。

sec_sess_stats.cur_rcv_pac_win
会话中 S U w = 窗 Z D 1 0 大小。

sec_sess_stats.send_data_frames
发 M D } # w }] 帧 D } ?。

sec_sess_stats.send_fmd_data_frames
发 M D } # w FMD }] 帧 D } ?。

sec_sess_stats.send_data_bytes
发 M D } # }] w D 字 Z }。

sec_sess_stats.rcv_data_frames
S U D } # w }] 帧 D } ?。

sec_sess_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧 D } ?。

sec_sess_stats.rcv_data_bytes
S U D } # }] w D 字 Z }。

sec_sess_stats.sidh
Session ID _ 字 Z。

sec_sess_stats.sidl
Session ID M 字 Z。

sec_sess_stats.odai

原 < 目 D X 址指 > 符。 1 启动会话 1， 如果 > X Z c | 含主 4 7 >， 则 BIND D发M方置C r 为0。如果 BIND D发M方G | 含从 4 7 > D Z c， 则置C r 为 1。

sec_sess_stats.ls_name

同统计信息相关 * D 4 7 > { F。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D。 C r 能C Z Q会话D统计信息通过信息 w ? 同 4 7 关 * 起来。

sec_sess_stats.pacing_type

在从会话中 9 C D S U同 = 类型。 C N } 能取 AP_NONE、AP_PACING_FIXED 或 AP_PACING_ADAPTIVE H值。

LOCAL_LU_INDICATION

无 [LOCAL LU 定义或 > } , z I C指 n 。 b 样允许注 a 过D & C L 序 # t 一份 1 0 定义D y P > X LU D P m。

VCB Structure

```
typedef struct local_lu_indication
{
   unsigned short  opcode;              /* verb operation code             */
   unsigned char   reserv2;             /* reserved                        */
   unsigned char   format;              /* format                          */
   unsigned short  primary_rc;          /* primary return code             */
   unsigned long   secondary_rc;        /* secondary return code           */
   unsigned char   data_lost;           /* previous indication lost        */
   unsigned char   reason;              /* reason for this indication      */
   unsigned char   lu_name[8];          /* LU name                         */
   unsigned char   description[RD_LEN];
                                        /* resource description            */
   unsigned char   lu_alias[8];         /* LU alias                        */
   unsigned char   nau_address;         /* NAU address                     */
   unsigned char   reserv4;             /* reserved                        */
   unsigned char   pu_name[8];          /* PU name                         */
   unsigned char   lu_sscp_active;      /* Is LU-SSCP session active       */
   unsigned char   reserv5;             /* reserved                        */
   SESSION_STATS   lu_sscp_stats;       /* LU-SSCP session statistics      */
   unsigned char   sscp_id[6];          /* SSCP ID                         */
} LOCAL_LU_INDICATION;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;        /* session receive RU size         */
   unsigned short  send_ru_size;       /* session send RU size            */
   unsigned short  max_send_btu_size;  /* max send BTU size               */
   unsigned short  max_rcv_btu_size;   /* max rcv BTU size                */
   unsigned short  max_send_pac_win;   /* max send pacing window size     */
   unsigned short  cur_send_pac_win;   /* current send pacing win size    */
   unsigned short  max_rcv_pac_win;    /* max receive pacing win size     */
   unsigned short  cur_rcv_pac_win;    /* curr receive pacing winsize     */
   unsigned long   send_data_frames;   /* number of data frames sent      */
   unsigned long   send_fmd_data_frames;
                                       /* num of FMD data frames sent     */
   unsigned long   send_data_bytes;    /* number of data bytes sent       */
   unsigned long   rcv_data_frames;    /* num of data frames received     */
   unsigned long   rcv_fmd_data_frames;
                                       /* num FMD data frames received    */
   unsigned long   rcv_data_bytes;     /* number of data bytes received   */
   unsigned char   sidh;               /* session ID high byte            */
   unsigned char   sidl;               /* session ID low byte             */
   unsigned char   odai;               /* ODAI bit set                    */
   unsigned char   ls_name[8];         /* Link station name               */
   unsigned char   pacing_type;        /* type of pacing in use           */
} SESSION_STATS;
```

": LU-SSCP 统计信息只P在1 **nau_address** 为非c R LU-SSCP 会话从活动D转为；活动D 1 P效。在y P其{ D i v下 # t C r 。

Parameters

opcode

AP_LOCAL_LU_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c , 以指定O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如 **data_lost** j 志位置为 AP_YES, 那 4 f 后 }] r 能置为U值。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

reason

发 v 指 n D原因:

AP_ADDED

LU 已定义。

AP_REMOVED

LU 已 > } , 要 4 w确 9 C DELETE_LOCAL_LU 或要 4 ; w确 9 C DELETE_LS、DELETE_PORT 或 DELETE_DLC。

AP_SSCP_ACTIVE

1 Z c I 功处理K ACTLU 后 LU-SSCP 会话处Z活动D状 , 。

AP_SSCP_INACTIVE

1 普通 DACTLU 或 4 7 ' \ 后 LU-SSCP 会话处Z ; 活动D状 , 。

lu_name

LU D { 字。状 , 已D d D > X LU。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ） , I EBCDIC U q R n d 。

description

资源 5 w（如 DEFINE_LOCAL_LU 中 5 w D ）。

lu_alias

> X定义D LU p { 。 b G v 以 I 显 > D > X字符集 m > D 8 字Z D字符串。 y P D 8 v字Z G P 效D。

nau_address

范围&在 0-255 之间D LU D网 g I 访问%元X址。非 c 值 | 含D LU G关 * LU。 c 值 | 含D LU G独 " LU。

pu_name

LU 9 C D PU D { 字。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串。如果 LU G关 * LU（即, **nau_address** 非 c ） , C r v在 b 1 P效, R 对Z独 " LU C r + 置为全 ? 二 x 制 c 。

lu_sscp_sess_active

5 w LU-SSCP 会话G否G活动D（AP_YES 或 AP_NO）。如果 **nau_address** 为 c , 那 4 # t C r 。

lu_sscp_stats.rcv_ru_size

< 终 # t C字段。

LOCAL_LU__INDICATION

lu_sscp_stats.send_ru_size
< 终 # t C字段。

lu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

lu_sscp_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

lu_sscp_stats.max_send_pac_win
< 终 + C字段 h 置为 c 。

lu_sscp_stats.cur_send_pac_win
< 终 + C字段 h 置为 c 。

lu_sscp_stats.max_rcv_pac_win
< 终 + C字段 h 置为 c 。

lu_sscp_stats.cur_rcv_pac_win
< 终 + C字段 h 置为 c 。

lu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ? 。

lu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ? 。

lu_sscp_stats.send_data_bytes
发M D } # }] w D字 Z } 。

lu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ? 。

lu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ? 。

lu_sscp_stats.rcv_data_bytes
S U D } # }] w D字 Z } 。

lu_sscp_stats.sidh
Session ID _ 字 Z 。

lu_sscp_stats.sidl
Session ID M字 Z 。

lu_sscp_stats.odai
原 < 目D X址指 > 符。 1 (" 会话 1 , 如果 > X Z c | 含主 4 7 > , 则 ACTLU D发M方置C r 为0; " R 如果 ACTLU D发M方 | 含从 4 7 > , 则置C r 为1。

lu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v字 Z G P 效D。 C r 能C Z Q会话D统计信息通过信息 w ? 同 4 7 关 * 起来。

lu_sscp_stats.pacing_type
LU-SSCP 会话中 9 C D S U同 = 类型。 C N } + 取 AP_NONE b v 值。

sscp_id
b G v | 含从C LU 9 C PU D ACTPU 中 S U D SSCP ID D6字 Z D r 。

Ｃｒｖｉ从ｔ LU ９Ｃ，Ｒ对Ｚ独" LU ＋置为全？Ｄ二ｘ制ｃ，或＿如果 **lu_sscp_sess_active** ；置为 AP_YES。

LOCAL_TOPOLOGY_INDICATION

1 在网 g Z c D拓扑 a 构 }] b D TG 项从活动D D d 为；活动D，或从活动D D d 为活动D 1 z I C指 n 。

VCB Structure

```
typedef struct local_topology_indication
{
   unsigned short  opcode;          /* verb operation code           */
   unsigned char   reserv2;         /* reserved                      */
   unsigned char   format;          /* format                        */
   unsigned short  primary_rc;      /* primary return code           */
   unsigned long   secondary_rc;    /* secondary return code         */
   unsigned char   data_lost;       /* previous indication lost      */
   unsigned char   status;          /* TG status                     */
   unsigned char   dest[17];        /* name of TG destination node   */
   unsigned char   dest_type;       /* TG destination node type      */
   unsigned char   tg_num;          /* TG number                     */
   unsigned char   cp_cp_session_active;
                                    /* CP-CP session is active       */
   unsigned char   branch_link_type;
                                    /* branch link type              */
   unsigned char   branch_tg;       /* TG is a branch TG             */
   unsigned char   reserva[17];     /* reserved                      */
} LOCAL_TOPOLOGY_INDICATION;
```

Parameters

opcode

AP_LOCAL_TOPOLOGY_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为U。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

status 指定 TG D状，。 C N } 能够C下 P 一 v 或多 v 一起做或运 c D值:

AP_TG_OPERATIVE
AP_TG_CP_CP_SESSIONS
AP_TG_QUIESCING
AP_NONE

dest TG D完 { 7 6目D Z c { 。 { F $ 度为 17 v字Z， " 以 EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（? v { F D $ 度最多 I 为 8 v字Z， R ；含 6 入U q 。）

dest_type

Z c D类型。 | G下 P 值之一:

AP_END_NODE
AP_NETWORK_NODE
AP_VRN

tg_num

k TG 相关D号 k 。

cp_cp_session_active

指定 > X Z c D : y $ 方 CP-CP 会话G否激活。

branch_link_type

v BrNN。 C TG 分支 4 7 类型。 C 值置为下 P 之一:

AP_UPLINK

C 4 7 G一种O行 4 7 。

AP_DOWNLINK

C 4 7 G一种 = EN D下行 4 7 。

AP_DOWNLINK_TO_BRNN

TG G一种显 > | D EN S Z D = BrNN D下行 4 7 。

AP_OTHERLINK

C 4 7 G一种其 | 4 7 。

其 | Z c 类型: C r 无意义R总G置为 AP_BRNN_NOT_SUPPORTED。

branch_tg

v NN。 j 6 TG G否G分支 TG。

AP_NO

TG ; G分支 TG。

AP_YES

TG G分支 TG。

其 | Z c 类型: C r ; P 意义R总置为 AP_NO。

LS_INDICATION

１９Ｃ４７Ｄ活动会话｝发ｚＤｄ，或４７＞Ｄ外？状，发ｚＤｄ，则ｚＩＣ指ｎ。４７＞Ｄ统计信息在４７＞ｄ为活动Ｄ１ａ供。

VCB Structure

```
typedef struct ls_indication
{
   unsigned short  opcode;              /* verb operation code           */
   unsigned char   attributes;          /* verb attributes               */
   unsigned char   reserv2;             /* reserved                      */
   unsigned char   format;              /* format                        */
   unsigned short  primary_rc;          /* primary return code           */
   unsigned long   secondary_rc;        /* secondary return code         */
   unsigned char   data_lost;           /* previous indication lost      */
   unsigned char   deactivated;         /* has session been deactivated? */
   unsigned char   ls_name[8];          /* link station name             */
   unsigned char   description[RD_LEN]; /* resource description          */
   unsigned char   adj_cp_name[17];     /* network qualified Adj CP name */
   unsigned char   adj_node_type;       /* adjacent node type            */
   unsigned short  act_sess_count;      /* active session count on link  */
   unsigned char   indication_cause;    /* cause of indication           */
   LS_STATS        ls_stats;            /* link station statistics       */
   unsigned char   tg_num;              /* TG number                     */
   unsigned long   sense_data;          /* sense data                    */
   unsigned char   brnn_link_type;      /* branch link type              */
   unsigned char   adj_cp_is_brnn;      /* adjacent CP is a BrNN         */
   unsigned char   reserva[17];         /* reserved                      */
} LS_INDICATION;

typedef struct ls_stats
{
   unsigned long   in_xid_bytes;        /* num of XID bytes received     */
   unsigned long   in_msg_bytes;        /* num message bytes received    */
   unsigned long   in_xid_frames;       /* num XID frames received       */
   unsigned long   in_msg_frames;       /* num message frames received   */
   unsigned long   out_xid_bytes;       /* num XID bytes sent            */
   unsigned long   out_msg_bytes;       /* num message bytes sent        */
   unsigned long   out_xid_frames;      /* number of XID frames sent     */
   unsigned long   out_msg_frames;      /* num message frames sent       */
   unsigned long   in_invalid_sna_frames;
                                        /* num invalid frames recvd      */
   unsigned long   in_session_control_frames;
                                        /* number of control             */
                                        /* frames recvd                  */
   unsigned long   out_session_control_frames;
                                        /* number of control             */
                                        /* frames sent                   */
   unsigned long   echo_rsps;           /* response from adj LS count    */
   unsigned long   current_delay;       /* time taken for last           */
                                        /* test signal                   */
   unsigned long   max_delay;           /* max delay by test signal      */
   unsigned long   min_delay;           /* min delay by test signal      */
   unsigned long   max_delay_time;      /* time since longest delay      */
   unsigned long   good_xids;           /* successful XID on LS count    */
   unsigned long   bad_xids;            /* unsuccessful XID on LS count  */
} LS_STATS;
```

Parameters

opcode

AP_LS_INDICATION

attributes

动词D t 性。 C字段G位字段。 Z一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

secondary_rc

H Z c。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1内？组件检b = 一次引起0一u指n丢 ' D故O后h置。如果 **data_lost** j志位置为 AP_YES，那4 f后D }] r能置为U。 C & C L序&发v一u QUERY | n C以 | 新已丢 ' D信息。

deactivated

1 LS d为活动D 1置为 AP_YES。 1 LS d为；活动D 1置为 AP_NO。

ls_name

4 7 > D { 字。 b G v以I显 > D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。

description

资源5 w（如在 DEFINE_LS 中5 w D）。 b G v以I显 > D > X字符集m > D 16 字Z D字符串。 y P D 16 v字Z G P效D。

adj_cp_name

完 { 7 6，17 字Z $，相Z X制c D { 字。 | I = v A 型 EBCDIC 字符串组I EBCDIC c串 *， R I EBCDIC U q R n d。（？v { F D $度最多I为8 v字Z， R；含6入U q。）

adj_node_type

Z c D类型。 | G下P值之一:

AP_END_NODE
AP_NETWORK_NODE
AP_LEN_NODE
AP_VRN

act_sess_count

9 C 4 7 D活动会话总？（端c和= i）。

indication_cause

C指n D原因。 | G下P值之一:

AP_ACTIVATION_STARTED
4 7 启动。

AP_ACTIVATING
4 7 d为活动D。

LS_INDICATION

AP_DEACTIVATION_STARTED
4 7 M放。

AP_DEACTIVATING
4 7 d为；活动D。

AP_SESS_COUNT_CHANGING
已D d D 9 C 4 7 D活动会话 } 目。

AP_CP_NAME_CHANGING
已D d其X制 c { 字D相Z Z c。

AP_FAILED
4 7 ' \。

AP_ACTIVATION_FAILED
4 7无法激活。

AP_PENDING_RETRY
已启动D重发定 1 器。 1 定 1 器， 1， 启C D 4 7自动重发。

AP_DATA_LOST
已丢 ' D 0 一 u 指 n。注意 4 7 > D统计信息只在 4 7 > 从活动D d 为；活动D 1 a 供（即, M放置为 AP_YES R **indication_cause** 置为 AP_DEACTIVATING）。在 y P 其 { D i v 下 # t C r。

ls_stats.in_xid_bytes
C 4 7 > S U D XID（；换 j 6 ）D字Z总？。

ls_stats.in_msg_bytes
C 4 7 > S U D }] 字Z总？。

ls_stats.in_xid_frames
C 4 7 > S U D XID（；换 j 6 ）D总？。

ls_stats.in_msg_frames
C 4 7 > S U D }] 帧D总？。

ls_stats.out_xid_bytes
C 4 7 > 发M D XID（；换 j 6 ）D总字Z }。

ls_stats.out_msg_bytes
C 4 7 > 发M D }] D总字Z }。

ls_stats.out_xid_frames
C 4 7 > 发M D XID（；换 j 6 ）帧D总？。

ls_stats.out_msg_frames
C 4 7 > 发M D }] 帧D总？。

ls_stats.in_invalid_sna_frames
C 4 7 > S U D ; } 确D SNA 帧D总？。

ls_stats.in_session_control_frames
C 4 7 > S U D会话X制帧D总？。

ls_stats.out_session_control_frames
C 4 7 > 发M D会话X制帧D总？。

ls_stats.echo_rsps

从相Z Z c S U D＆答。周期性D发M＆答k s以规定=相Z Z c D传%延1。

ls_stats.current_delay

从b v 4 7 > = Z S 4 7 >发M和返回D最Y D b T信号y花D 1间（毫k）。

ls_stats.max_delay

从C 4 7 > = Z S 4 7 >发M和返回b T信号y花D最$ 1间（毫k）。

ls_stats.min_delay

从C 4 7 > = Z S 4 7 >发M和返回b T信号y花D最短1间（毫k）。

ls_stats.max_delay_time

1延Y发z 1，从系统启动* < D 1间（以Y分之一k为%位）。

ls_stats.good_xids

从4 7 >启动起，发z在4 7 > O D I功D XID ；换D总？。

ls_stats.bad_xids

从4 7 >启动起，发z在4 7 > O D ； I功D XID ；换D总？。

tg_num

k TG 相关D号k。

sense_data

如v人通信或通信服务器检b }] 置位，检b = 一v XID 协议错误。} 非**indication_cause** 为 AP_FAILED，则# t C r。

brnn_link_type

v BrNN。C分支4 7类型，｜G下P之一:

AP_UPLINK

LU_0_TO_3_INDICATION

1 > X LU（类型0-3）D状，d化1，发v C指n。

VCB Structure

```
typedef struct lu_0_to_3_indication
{
   unsigned short  opcode;              /* verb operation code          */
   unsigned char   attributes;          /* attributes                   */
   unsigned char   reserv2;             /* reserved                     */
   unsigned char   format;              /* format                       */
   unsigned short  primary_rc;          /* primary return code          */
   unsigned long   secondary_rc;        /* secondary return code        */
   unsigned char   data_lost;           /* previous indication lost     */
   unsigned char   pu_name[8];          /* PU Name                      */
   unsigned char   lu_name[8];          /* LU Name                      */
   unsigned char   description[RD_LEN]; /* resource description         */
   unsigned char   nau_address;         /* NAU address                  */
   unsigned char   lu_sscp_sess_active;
                                        /* Is SSCP session active?      */
   unsigned char   appl_conn_active;    /* Is application using LU?     */
   unsigned char   plu_sess_active;     /* Is PLU-SLU session active?   */
   unsigned char   host_attachment;     /* Host attachment              */
   SESSION_STATS   lu_sscp_stats;       /* LU-SSCP session statistics   */
   SESSION_STATS   plu_stats;           /* PLU-SLU session statistics   */
   unsigned char   sscp_id[16];         /* SSCP ID                      */
} LU_0_TO_3_INDICATION;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;        /* session receive RU size       */
   unsigned short  send_ru_size;       /* session send RU size          */
   unsigned short  max_send_btu_size;  /* max send BTU size             */
   unsigned short  max_rcv_btu_size;   /* max rcv BTU size              */
   unsigned short  max_send_pac_win;   /* max send pacing window size   */
   unsigned short  cur_send_pac_win;   /* current send pacing win size  */
   unsigned short  max_rcv_pac_win;    /* max receive pacing win size   */
   unsigned short  cur_rcv_pac_win;    /* curr receive pacing winsize   */
   unsigned long   send_data_frames;   /* number of data frames sent    */
   unsigned long   send_fmd_data_frames;
                                       /* num of FMD data frames sent   */
   unsigned long   send_data_bytes;    /* number of data bytes sent     */
   unsigned long   rcv_data_frames;    /* num of data frames received   */
   unsigned long   rcv_fmd_data_frames;
                                       /* num FMD data frames received  */
   unsigned long   rcv_data_bytes;     /* number of data bytes received */
   unsigned char   sidh;               /* session ID high byte          */
   unsigned char   sidl;               /* session ID low byte           */
   unsigned char   odai;               /* ODAI bit set                  */
   unsigned char   ls_name[8];         /* Link station name             */
   unsigned char   pacing_type;        /* type of pacing in use         */
} SESSION_STATS;
```

Parameters

opcode

AP_LU_0_TO_3_INDICATION

attributes

动词D t 性。C字段G位字段。Z一位 | 含要定义资源D I 见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

如 }] 丢 ' 置为 AP_YES, b 儿置为 AP_EXTERNALLY_VISIBLE。

format

j 6 VCB D q = 。 + C字段 h 置为 c , 以指定 O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故 O后 h 置。如果 **data_lost** j 志位置为 AP_YES, 那 4 f 后 D }] r 能置为U。 C & C L 序 & 发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

pu_name

> X PU D { 字。 b G v 8 字 Z D字母 } 字集 D A 型 EBCDIC 字符串（以一 v 字母 * < ）, I EBCDIC U q R n d 。

lu_name

状 , 已 D d D > X LU。 b G v 8 字 Z D字母 } 字集 D A 型 EBCDIC 字符串（以一 v 字母 * < ）, I EBCDIC U q R n d 。

description

资源 5 w（如在 DEFINE_LU_0_TO_3 中 5 w D ）。 b G v 以 I 显 > D > X字符集 m > D 16 字 Z D字符串。 y P D 16 v 字 Z G P 效 D 。

nau_address

LU D网 g I 访问 % 元 X 址（范围 & 在10--2554之间 ）。

lu_sscp_sess_active

j 6 ACTLU G否已 I 功处理（AP_YES 或 AP_NO ）。

appl_conn_active

9 C C LU 1 置位（AP_YES 或 AP_NO ）。

plu_sess_active

5 w PLU-SLU 会话 G 否已激活（AP_YES 或 AP_NO ）。

host_attachment

指定 LU 主机 D , S 类型:

AP_DLUR_ATTACHED

LU 9 C DLUR 同主机系统相 , 。

AP_DIRECT_ATTACHED

LU 同主机系统直 S , S 。注意 LU-SSCP 和 PLU-SLU D统计信息只在会话从激活 = 非激活 D i v 下 P 效。在 y P 其 { D i v 下 # t C r 。

lu_sscp_stats.rcv_ru_size

< 终 # t C字段。

LU_0_TO_3__INDICATION

lu_sscp_stats.send_ru_size
< 终 # t C 字段。

lu_sscp_stats.max_send_btu_size
能发M D 最大 BTU 大小。

lu_sscp_stats.max_rcv_btu_size
能 S U D 最大 BTU 大小。

lu_sscp_stats.max_send_pac_win
< 终 + C 字段 h 置为 c 。

lu_sscp_stats.cur_send_pac_win
< 终 + C 字段 h 置为 c 。

lu_sscp_stats.max_rcv_pac_win
< 终 + C 字段 h 置为 c 。

lu_sscp_stats.cur_rcv_pac_win
< 终 + C 字段 h 置为 c 。

lu_sscp_stats.send_data_frames
发M D } # w }] 帧 D } ? 。

lu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧 D } ? 。

lu_sscp_stats.send_data_bytes
发M D } # }] w D 字 Z } 。

lu_sscp_stats.rcv_data_frames
S U D } # w }] 帧 D } ? 。

lu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧 D } ? 。

lu_sscp_stats.rcv_data_bytes
S U D } # }] w D 字 Z } 。

lu_sscp_stats.sidh
Session ID _ 字 Z 。

lu_sscp_stats.sidl
Session ID M 字 Z 。

lu_sscp_stats.odai
原 < 目 D X 址指 > 符。 1 (" 会话 1 , 如果 > X Z c | 含主 4 7 > , 则 ACTLU D 发M方置C r 为0; " R 如果 ACTLU D 发M方 | 含从 4 7 > , 则置C r 为1。

lu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 y P D 8 v 字 Z G P 效 D 。 C r 能 C Z Q 会话 D 统计信息通过信息 w ? 同 4 7 关 * 起来。

lu_sscp_stats.pacing_type
LU-SSCP 会话中 9 C D S U 同 = 类型。 C N } + 取 AP_NONE b v 值。

plu_stats.rcv_ru_size
最大 S U RU 大小。

plu_stats.send_ru_size
最大发M RU 大小。

plu_stats.max_send_btu_size
能发M D最大 BTU 大小。

plu_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

plu_stats.max_send_pac_win
会话中发M w = 窗 Z D最大 _ 寸。

plu_stats.cur_send_pac_win
会话中发M w = 窗 Z D 1 0大小。

plu_stats.max_rcv_pac_win
会话中 S U w = 窗 Z D最大 _ 寸。

plu_stats.cur_rcv_pac_win
会话中 S U w = 窗 Z D 1 0大小。

plu_stats.send_data_frames
发M D } # w }] 帧D } ?。

plu_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ?。

plu_stats.send_data_bytes
发M D } # }] w D字 Z }。

plu_stats.rcv_data_frames
S U D } # w }] 帧D } ?。

plu_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ?。

plu_stats.rcv_data_bytes
S U D } # }] w D字 Z }。

plu_stats.sidh
Session ID _ 字 Z。

plu_stats.sidl
Session ID M字 Z。

plu_stats.odai
原 < 目 D X址指 > 符。 1 (" 会话 1， 如果 > X Z c | 含主 4 7 >， 则 ACTLU D发M方置C r 为0; " R如果 ACTLU D发M方 | 含从 4 7 >， 则置C r 为1。

plu_stats.ls_name
同统计信息相关 * D 4 7 > { F。 b G v以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v字 Z G P效D。 C r 能C Z Q会话D统计信息通过信息 w ? 同 4 7 关 * 起来。

plu_stats.pacing_type
在 PLU-SLU 会话中 9 C D S U同 = 类型。 b + 取 AP_NONE 或 AP_PACING_FIXED H值。

sscp_id
b G v | 含从C LU 9 C PU D ACTPU 中 S U D SSCP ID D6字 Z D r 。

LU_0_TO_3__INDICATION

如果 **lu_sscp_sess_active** ；G AP_YES，则C r 置c 。

MODE_INDICATION

1 > X LU 和对方 LU * < 组合 9 C 一定 D 方 = 1，R 对 Z > X LU、对方 LU D 1 0 会话记 } 1， R 组合方 = D D d 1 发 v C 指 n 。

VCB Structure

```
typedef struct mode_indication
{
   unsigned short  opcode;               /* verb operation code       */
   unsigned char   reserv2;              /* reserved                  */
   unsigned char   format;               /* format                    */
   unsigned short  primary_rc;           /* primary return code       */
   unsigned long   secondary_rc;         /* secondary return code     */
   unsigned char   data_lost;            /* previous indication lost  */
   unsigned char   removed;              /* is entry being removed?   */
   unsigned char   lu_alias[8];          /* LU alias                  */
   unsigned char   plu_alias[8];         /* partner LU alias          */
   unsigned char   fqplu_name[17];       /* fully qualified partner   */
                                         /* LU name                   */
   unsigned char   mode_name[8];         /* mode name                 */
   unsigned char   description[RD_LEN];  /* resource description      */
   unsigned short  curr_sess_count;      /* current session count     */
   unsigned char   reserva[20];          /* reserved                  */
} MODE_INDICATION;
```

Parameters

opcode

AP_MODE_INDICATION

format

j 6 VCB D q = 。 + C 字段 h 置为 c，以指定 O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G 否已 - 丢 '（AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D 故 O 后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为 U。 C & C L 序 & 发 v 一 u QUERY | n C 以 | 新已丢 ' D 信息。

removed

5 w 一项 G 否已 > }（AP_YES 或 AP_NO）。 1 > } 项，过增加项 1 置位。

lu_alias

> X 定义 D LU p { 。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 y P D 8 v 字 Z G P 效 D。

plu_alias

对方 LU D p { 。 b G v 以 I 显 > D > X 字符集 m > D 8 字 Z D 字符串。 y P D 8 v 字 Z G P 效 D。

fqplu_name

C Z 伙 i LU D 17 字 Z 完 { 7 6 网 g { 。 C { F G I = v C EBCDIC c 串

＊D A 型 EBCDIC 字符串构I， " C EBCDIC U q R n d。（？v { F D $ 度最多I为 8 v字Z， R ；含6入U q。）

mode_name

为一组会话指定网g t 性D方 = { 。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。

description

资源5 w（如在 DEFINE_MODE 中5 w D ）。 b G v以I显 > D > X字符集m > D 16 字Z D字符串。 y P D 16 v字Z G P效D。

curr_sess_count

对Z C > X LU，对方 LU，组合方 = D会话 } 。

NN_TOPOLOGY_NODE_INDICATION

C动词ｖＪＣＺ通信服务器。

１在网ｇＺｃＤ拓扑ａ构｝］ｂ中ＤＺｃ项从活动ＤＤｄ为；活动Ｄ，或从活动ＤＤｄ为活动Ｄ１ｚＩＣ指ｎ。

VCB Structure

```
typedef struct nn_topology_node_indication
{
   unsigned short  opcode;         /* verb operation code           */
   unsigned char   reserv2;        /* reserved                      */
   unsigned char   format;         /* format                        */
   unsigned short  primary_rc;     /* primary return code           */
   unsigned long   secondary_rc;   /* secondary return code         */
   unsigned char   data_lost;      /* previous indication lost      */
   unsigned char   deactivated;    /* has the node become inactive? */
   unsigned char   node_name[17];  /* node name                     */
   unsigned char   node_type;      /* node type                     */
   unsigned char   branch_aware;   /* node is branch aware          */
   unsigned char   reserva[19];    /* reserved                      */
} NN_TOPOLOGY_NODE_INDICATION;
```

Parameters

opcode

AP_NN_TOPOLOGY_TG_INDICATION

format

ｊ６ VCB Ｄｑ＝。＋C字段ｈ置为ｃ，以指定Ｏｆ Ｐｖ Ｄ VCB Ｄｆ＞。

primary_rc

AP_OK

data_lost

５ｗ｝］G否已－丢＇（AP_YES 或 AP_NO）。１内？组件检ｂ＝一次引起０一ｕ指ｎ丢＇Ｄ故Ｏ后ｈ置。如果 **data_lost** ｊ志位置为 AP_YES，那４ｆ后Ｄ｝］ｒ能置为Ｕ。Ｃ＆ＣＬ序＆发ｖ一ｕ QUERY ｜ｎC以｜新已丢＇Ｄ信息。

deactivated

１Ｚｃ处Ｚ非激活状，１置为 AP_YES。１Ｚｃ处Ｚ激活状，１置为 AP_YES。

node_name

来自网ｇ拓扑｝］ｂＤ网ｇ完｛７６Ｚｃ｛Ｆ。｛Ｆ＄度为 17 ｖ字Ｚ，＂以 EBCDIC Ｕｑ Ｒｎｄ。｜Ｉ＝ｖ Ａ 型 EBCDIC 字符串组Ｉ，中间以一ｖ EBCDIC ｃ，Ｓ。（？ｖ｛ＦＤ＄度最多Ｉ为 ８ ｖ字Ｚ， Ｒ；含６入Ｕｑ。）

node_type

Ｚｃ Ｄ类型。｜Ｇ下Ｐ之一:

AP_NETWORK_NODE
AP_VRN

NN_TOPOLOGY_NODE__INDICATION

branch_aware

5 w Z c G否G分支。

AP_NO

C Z c ；G分支。

AP_YES

C Z c G分支。

NN_TOPOLOGY_TG_INDICATION

C动词v J C Z通信服务器。

1在网g Z c D拓扑a构}] b D TG 项从活动D D d为；活动D，或从活动D D d为活动D 1 z I C指n。

VCB Structure

```
typedef struct nn_topology_tg_indication
{
   unsigned short  opcode;         /* verb operation code           */
   unsigned char   reserv2;        /* reserved                      */
   unsigned char   format;         /* format                        */
   unsigned short  primary_rc;     /* primary return code           */
   unsigned long   secondary_rc;   /* secondary return code         */
   unsigned char   data_lost;      /* previous indication lost      */
   unsigned char   status;         /* TG status                     */
   unsigned char   owner[17];      /* name of TG owner node         */
   unsigned char   dest[17];       /* name of TG destination node   */
   unsigned char   tg_num;         /* TG number                     */
   unsigned char   owner_type;     /* Type of node that owns the TG */
   unsigned char   dest_type;      /* TG destination node type      */
   unsigned char   cp_cp_session_active;
                                   /* CP-CP session is active       */
   unsigned char   branch_tg;      /* TG is a branch TG             */
   unsigned char   reserva[16];    /* reserved                      */
} NN_TOPOLOGY_TG_INDICATION;
```

Parameters

opcode

AP_NN_TOPOLOGY_TG_INDICATION

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1内？组件检b = 一次引起0一u指n丢 ' D故O后h置。如果 **data_lost** j志位置为 AP_YES，那4 f后D }] r能置为U。 C & C L序&发v一u QUERY | n C以 | 新已丢 ' D信息。

status 指定 TG D状，。 C N } 能够C下P一v或多v和 ORed 一起D值:

AP_TG_OPERATIVE
AP_TG_QUIESCING
AP_TG_CP_CP_SESSIONS
AP_NONE

owner TG D发自Z c { 字（总置为> X Z c { ）。 { F $度为 17 v字Z， " 以EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， R ；含6入U q。）

NN_TOPOLOGY_TG_INDICATION

dest TG D完{ 7 6目D Z c { 。 { F $度为 17 v字Z， "以 EBCDIC U q R n d 。 | I = v A 型 EBCDIC 字符串组I，中间以一v EBCDIC c ，S。（？v { F D $度最多I为 8 v字Z， R ；含6人U q 。）

tg_num
k TG 相关D号k。

owner_type
5 P TG D Z c类型。

AP_NETWORK_NODE
AP_VRN

dest_type
Z c D类型。

AP_NETWORK_NODE
AP_VRN

cp_cp_session_active
指定5 P Z c D： y $方 CP-CP 会话G否激活（AP_NO 或 AP_YES）。

branch_tg
j 6 TG G否G分支 TG。

AP_NO
TG ；G分支 TG。

AP_YES
TG G分支 TG。

PLU_INDICATION

1 > X LU W先同对方 LU ，S 1 z I C指n。 1处理x C PLU D Z一v ALLOCATE或从C PLU S U Z一v BIND 1 +会发z 。如对方X制c D { 字D d，总G z I C指n。

VCB Structure

```
typedef struct plu_indication
{
   unsigned short  opcode;              /* verb operation code        */
   unsigned char   reserv2;             /* reserved                   */
   unsigned char   format;              /* format                     */
   unsigned short  primary_rc;          /* primary return code        */
   unsigned long   secondary_rc;        /* secondary return code      */
   unsigned char   data_lost;           /* has previous indication    */
                                        /* been lost?                 */
   unsigned char   removed;             /* is entry being removed?    */
   unsigned char   lu_alias[8];         /* LU alias                   */
   unsigned char   plu_alias[8];        /* partner LU alias           */
   unsigned char   fqplu_name[17];      /* fully qualified partner    */
                                        /* LU name                    */
   unsigned char   description[RD_LEN]; /* resource description       */
   unsigned char   partner_cp_name[17]; /* partner CP name            */
   unsigned char   partner_lu_located;  /* partner CP name resolved?  */
   unsigned char   reserva[20];         /* reserved                   */
} PLU_INDICATION;
```

Parameters

opcode

AP_PLU_INDICATION

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

secondary_rc

H Z c。

data_lost

5 w一v或多v指n G否已丢 '（AP_YES 或 AP_NO）。 1内？组件无法发M 0一u指n，则置位。如果 **data_lost** j志位置为 AP_YES，那4 f后D }] r能置为U。 C & C L序&发v一u QUERY | n C以 | 新已丢 ' D信息。

removed

5 w一项G否已 > }（AP_YES 或 AP_NO）。 1 > }项，过增加项1置位。

lu_alias

> X定义D LU p {。 b G v以I显> D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。

plu_alias

对方 LU D p {。 b G v以I显> D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。

PLU_INDICATION

fqplu_name

C Z 伙 i LU D 17 字 Z 完 { 7 6 网 g { 。 C { F I = v A-型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S， " 以 EBCDIC U q R n d。（ ? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q 。）

description

资源 5 w（如在 DEFINE_PARTNER_LU 中 5 w D)。 b G v 以 I 显 > D > X 字符集 m > D 16 字 Z D 字符串。 y P D 16 v 字 Z G P 效 D。

partner_cp_name

伙 i LU X 制 c D 17 字 Z 完 { 7 6 网 g { F。 C { F G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I， " C EBCDIC U q R n d。（ ? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q 。）

partner_lu_located

5 w 对方 X 制 c D { 字 G 否已分 b （AP_YES 或 AP_NO）， R C r **partner_cp_name** G 否 | 含 X 制 c { 字。

PORT_INDICATION

1端Z从活动D d为；活动D（或反之亦然）1， z I C指n。

VCB Structure

```
typedef struct port_indication
{
   unsigned short  opcode;               /* verb operation code           */
   unsigned char   reserv2;              /* reserved                      */
   unsigned char   attributes;           /* verb attributes               */
   unsigned char   format;               /* format                        */
   unsigned short  primary_rc;           /* primary return code           */
   unsigned long   secondary_rc;         /* secondary return code         */
   unsigned char   data_lost;            /* previous indication lost      */
   unsigned char   deactivated;          /* has session been deactivated? */
   unsigned char   port_name[8];         /* link station name             */
   unsigned char   description[RD_LEN];  /* resource description          */
   unsigned char   reserva[20];          /* reserved                      */
} PORT_INDICATION;
```

Parameters

opcode

AP_PORT_INDICATION

attributes

动词D t性。C字段G位字段。Z一位|含要定义资源D I见性，" k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

primary_rc

AP_OK

secondary_rc

H Z c。

data_lost

5 w }] G否已 - 丢 '（AP_YES 或 AP_NO）。1内？组件检b =一次引起0一u指n丢 ' D故O后h置。如果 **data_lost** j志位置为 AP_YES，那4 f后D }] r能置为U。C & C L序&发v一u QUERY | n C以|新已丢 ' D信息。

deactivated

1端Z处Z非激活状，1置为 AP_YES。1 Z c处Z激活状，1置为 AP_NO。

port_name

端Z D { 字。b G v以I显> D > X字符集m > D 8 字Z D字符串。y P D 8 v字Z G P效D。

description

资源5 w（如在 DEFINE_PORT 中5 w D）。b G v以I显> D > X字符集m > D 16 字Z D字符串。y P D 16 v字Z G P效D。

PU_INDICATION

1 > X LU D状，d化1， z I C指n。

VCB Structure

```
typedef struct pu_indication
{
   unsigned short  opcode;               /* verb operation code           */
   unsigned char   attributes;           /* attributes                    */
   unsigned char   reserv2;              /* reserved                      */
   unsigned char   format;               /* format                        */
   unsigned short  primary_rc;           /* primary return code           */
   unsigned long   secondary_rc;         /* secondary return code         */
   unsigned char   data_lost;            /* previous indication lost      */
   unsigned char   pu_name[8];           /* PU Name                       */
   unsigned char   description[RD_LEN];  /* resource description          */
   unsigned char   pu_sscp_sess_active;
                                         /* Is SSCP session active?       */
   unsigned char   host_attachment;      /* Host attachment               */
   unsigned char   reserv1[2];           /* reserved                      */
   SESSION_STATS   pu_sscp_stats;        /* PU-SSCP session statistics    */
   unsigned char   sscp_id[6];           /* SSCP ID                       */
} PU_INDICATION;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;          /* session receive RU size       */
   unsigned short  send_ru_size;         /* session send RU size          */
   unsigned short  max_send_btu_size;    /* max send BTU size             */
   unsigned short  max_rcv_btu_size;     /* max rcv BTU size              */
   unsigned short  max_send_pac_win;     /* max send pacing window size   */
   unsigned short  cur_send_pac_win;     /* curr send pacing window size  */
   unsigned short  max_rcv_pac_win;      /* max rcv pacing window size    */
   unsigned short  cur_rcv_pac_win;      /* curr receive pacing win size  */
   unsigned long   send_data_frames;     /* number of data frames sent    */
   unsigned long   send_fmd_data_frames;
                                         /* num FMD data frames sent      */
   unsigned long   send_data_bytes;      /* number of data bytes sent     */
   unsigned long   rcv_data_frames;      /* num of data frames received   */
   unsigned long   rcv_fmd_data_frames;
                                         /* num FMD data frames received  */
   unsigned long   rcv_data_bytes;       /* num data bytes received       */
   unsigned char   sidh;                 /* session ID high byte          */
   unsigned char   sidl;                 /* session ID low byte           */
   unsigned char   odai;                 /* ODAI bit set                  */
   unsigned char   ls_name[8];           /* Link station name             */
   unsigned char   pacing_type;          /* type of pacing in use         */
} SESSION_STATS;
```

Parameters

opcode

AP_PU_INDICATION

attributes

动词D t 性。C字段G位字段。Z一位 | 含要定义资源D I 见性， " k下P之一对&:

AP_EXTERNALLY_VISIBLE
AP_INTERNALLY_VISIBLE

如 }] 丢 ' 置为 AP_YES, b 儿置为 AP_EXTERNALLY_VISIBLE。

format
j 6 VCB D q = 。 + C字段 h 置为 c , 以指定O f P v D VCB D f > 。

primary_rc
AP_OK

secondary_rc
H Z c 。

data_lost
5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES, 那 4 f 后 D }] r 能置为U。 C & C L 序&发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

pu_name
PU D { 字（在 DEFINE_LS | n 中配置D）。 b G v 8 字 Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ）, I EBCDIC U q R n d 。

description
资源 5 w（如在 DEFINE_LS 或 DEFINE_INTERNAL_PU 中 5 w D ）。 b G v 以 I 显 > D > X字符集 m > D 16 字 Z D字符串。 y P D 16 v 字 Z G P 效D。

pu_sscp_sess_active
j 6 ACTPU G否已 I 功处理（AP_YES 或 AP_NO）。

host_attachment
PU 主机 , S 类型:

AP_DLUR_ATTACHED
PU 9 C DLUR 同主机系统 , S 。

AP_DIRECT_ATTACHED
PU 同主机系统直 S , S 。

" : PU-SSCP 统计信息只在会话状 , 从激活 = 非激活 1 P 效。
在 y P i v 下 # t 以下 r :

pu_sscp_stats.rcv_ru_size
< 终 # t C字段。

pu_sscp_stats.send_ru_size
< 终 # t C字段。

pu_sscp_stats.max_send_btu_size
能发M D最大 BTU 大小。

pu_sscp_stats.max_rcv_btu_size
能 S U D最大 BTU 大小。

pu_sscp_stats.max_send_pac_win
< 终 + C字段 h 置为 c 。

pu_sscp_stats.cur_send_pac_win
< 终 + C字段 h 置为 c 。

pu_sscp_stats.max_rcv_pac_win
< 终 + C 字段 h 置为 c 。

pu_sscp_stats.cur_rcv_pac_win
< 终 + C 字段 h 置为 c 。

pu_sscp_stats.send_data_frames
发M D } # w }] 帧D } ? 。

pu_sscp_stats.send_fmd_data_frames
发M D } # w FMD }] 帧D } ? 。

pu_sscp_stats.send_data_bytes
发M D } # }] w D字 Z } 。

pu_sscp_stats.rcv_data_frames
S U D } # w }] 帧D } ? 。

pu_sscp_stats.rcv_fmd_data_frames
S U D } # w FMD }] 帧D } ? 。

pu_sscp_stats.rcv_data_bytes
S U D } # }] w D字 Z } 。

pu_sscp_stats.sidh
Session ID _ 字 Z 。

pu_sscp_stats.sidl
Session ID M字 Z 。

pu_sscp_stats.odai
原 < 目D X址指 > 符。 1 会话启动 1 , 如果C Z c | 含主 4 7 > , 则 ACTPU D 发M方置C r 为0; " R 如果 ACTPU G | 含从 4 7 > D Z c , 则置C r 为1。

pu_sscp_stats.ls_name
同统计信息相关 * D 4 7 > { F 。 b G v 以 I 显 > D > X字符集m > D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D。 C r 能C Z Q会话D统计信息通过信息 w ? 同 4 7 关 * 起来。

pu_stats.pacing_type
PU-SSCP 会话中 9 C D S U同 = 类型。 C N } + 取 AP_NONE b v 值。

sscp_id
b G v | 含从 PU D ACTPU 中S U D SSCP ID D 6 字Z D r 。

如 **plu_sscp_sess_active** ; G AP_YES, 然后C r + 置 c 。

REGISTER_INDICATION_SINK

REGISTER_INDICATION_SINK 注 a K & C发M x 一 v x L 和队 P D指 n 。

在 verb_signal W ? D任何 S U指 n 中返回 REGISTER_INDICATION_SINK D verb_signal W ? 中D **orig_verb_data** 。

VCB Structure

```
typedef struct port_indication
{
   unsigned short  opcode;               /* verb operation code          */
   unsigned char   reserv2;              /* reserved                     */
   unsigned char   format;               /* format                       */
   unsigned short  primary_rc;           /* primary return code          */
   unsigned long   secondary_rc;         /* secondary return code        */
   unsigned PROC_ID
                   proc_id;              /* process identifier of sink   */
   unsigned QUEUE_ID
                   queue_id;             /* queue identifier where       */
                                         /* indications will be sent     */
   unsigned short  indication_opcode;    /* opcode of indication to      */
                                         /* be sunk                      */
} REGISTER_INDICATION_SINK;
```

Parameters

opcode

AP_REGISTER_INDICATION_SINK

format

j 6 VCB D q = 。 + C字段 h 置为 c , 以指定O f P v D VCB D f > 。

proc_id

Process ID D S U处理。

queue_id

X需发M S U处理指 n D Queue ID。

indication_opcode

一) 指 n S U器注 a , 无 [何 1 z I , 都会返回指 n D Y作 k 。

返回N数

如果C动词 I 功执行, 则L序返回下 P N } :

primary_rc

AP_OK

如果C动词因 N } 错误而未能执行, 则L序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_OP_CODE

AP_DYNAMIC_LOAD_ALREADY_REGD

如果C动词因状，错误而未能执行，则L序返回下P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_INVALID_LU_NAME

如果C动词因 } 确D START_NODE N } ; P发M而未能执行，则L序返回下P N }:

primary_rc
AP_FUNCTION_NOT_SUPPORTED

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因 STOP_NODE | n P未发v而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

REGISTRATION_FAILURE

REGISTRATION_FAILURE 指 v K I 网 g Z c 服务器注 a 资源 D ' \ D 一次企图。

VCB Structure

```
typedef struct registration_failure
{
   unsigned short  opcode;               /* verb operation code      */
   unsigned char   reserv2;              /* reserved                 */
   unsigned char   format;               /* format                   */
   unsigned short  primary_rc;           /* primary return code      */
   unsigned long   secondary_rc;         /* secondary return code    */
   unsigned char   data_lost;            /* previous indication lost */
   unsigned char   resource_name[17];    /* network qualified        */
                                         /* resource name            */
   unsigned short  resource_type;        /* resource type            */
   unsigned char   description[RD_LEN];  /* resource description     */
   unsigned char   reserv2b[2];          /* reserved                 */
   unsigned long   sense_data;           /* sense data               */
   unsigned char   reserva[20];          /* reserved                 */
} REGISTRATION_FAILURE;
```

Parameters

opcode

AP_REGISTRATION_FAILURE

format

j 6 VCB D q = 。 + C 字段 h 置为 c，以指定 O f P v D VCB D f > 。

primary_rc

AP_OK

data_lost

5 w }] G 否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D 故 O 后 h 置。如 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为 U。 C & C L 序 & 发 v 一 u QUERY | n C 以 | 新已丢 ' D 信息。

resource_name

无法注 a D 资源 { 字。 { F $ 度为 17 v 字 Z， " 以 EBCDIC U q R n d 。 C { 字 G I = v C EBCDIC c 串 * D A 型 EBCDIC 字符串构 I 。（ ? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q 。）

resource_type

资源类型。 | G 下 P 之一:

AP_NNCP_RESOURCE
AP_ENCP_RESOURCE
AP_LU_RESOURCE

description

资源 5 w（如在 DEFINE_LOCAL_LU 或 DEFINE_ADJACENT_NODE 中 5 w D ）。

sense_data

检 b }]（在 *SNA q* = 中 5 w D ）。

RTP_INDICATION

1 :

- RTP ， S D ， S 或断 *
- 活动对话 x 计 } D D d
- ， S 执行 7 6 P 换。

1 ， S 断 * 1， + 会返回最后 D RTP 统计信息。在其 | 1 间 # t C r **rtp_stats** 。

VCB Structure

```
typedef struct rtp_indication
{
   unsigned short  opcode;              /* verb operation code            */
   unsigned char   reserv2;             /* reserved                       */
   unsigned char   format;              /* format                         */
   unsigned short  primary_rc;          /* primary return code            */
   unsigned long   secondary_rc;        /* secondary return code          */
   unsigned char   data_lost;           /* previous indication(s) lost    */
   unsigned char   connection_state;    /* the current state of the RTP   */
                                        /* connection                     */
   unsigned char   rtp_name[8];         /* name of the RTP connection     */
   unsigned short  num_sess_active;     /* number of active sessions      */
   unsigned char   indication_cause;    /* reason for this indication     */
   unsigned char   reserv3[3];          /* reserved                       */
   RTP_STATISTICS  rtp_stats;           /* RTP statistics                 */
} RTP_INDICATION;

 typedef struct rtp_statistics
{
   unsigned long   bytes_sent;          /* total num of bytes sent        */
   unsigned long   bytes_received;      /* total num bytes received       */
   unsigned long   bytes_resent;        /* total num of bytes resent      */
   unsigned long   bytes_discarded;     /* total num bytes discarded      */
   unsigned long   packets_sent;        /* total num of packets sent      */
   unsigned long   packets_received;    /* total num packets received     */
   unsigned long   packets_resent;      /* total num of packets resent    */
   unsigned long   packets_discarded;   /* total num packets discarded    */
   unsigned long   gaps_detected;       /* gaps detected                  */
   unsigned long   send_rate;           /* current send rate              */
   unsigned long   max_send_rate;       /* maximum send rate              */
   unsigned long   min_send_rate;       /* minimum send rate              */
   unsigned long   receive_rate;        /* current receive rate           */
   unsigned long   max_receive_rate;    /* maximum receive rate           */
   unsigned long   min_receive_rate;    /* minimum receive rate           */
   unsigned long   burst_size;          /* current burst size             */
   unsigned long   up_time;             /* total uptime of connection     */
   unsigned long   smooth_rtt;          /* smoothed round-trip time       */
   unsigned long   last_rtt;            /* last round-trip time           */
   unsigned long   short_req_timer;     /* SHORT_REQ timer duration       */
   unsigned long   short_req_timeouts;  /* number of SHORT_REQ timeouts   */
   unsigned long   liveness_timeouts;   /* number of liveness timeouts    */
   unsigned long   in_invalid_sna_frames;
                                        /* number of invalid SNA frames   */
                                        /* received                       */
   unsigned long   in_sc_frames;        /* number of SC frames received   */
   unsigned long   out_sc_frames;       /* number of SC frames sent       */
   unsigned char   reserve[40];         /* reserved                       */
} RTP_STATISTICS;
```

Parameters

opcode

AP_RTP_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c , 以指定O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G否已 - 丢 ' （AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如 **data_lost** j 志位置为 AP_YES 那 4 从 0 一 u 指 n S U 1 * <, | 含D }] I 能D d K ; 止一次。

connection_state

RTP , S D 1 0 状 , 。 | G下 P 值之一:

AP_CONNECTING

, S h 置已 - * <, + 还 ; P a x 。

AP_CONNECTED

, S 完全 P 效。

AP_DISCONNECTED

, S ; 再 P 效。

rtp_name

RTP , S D { 字。 b G v 以 I 显 > D > X字符集 m > D 8 字 Z D字符串。 y P D 8 v 字 Z G P 效D。

num_sess_active

在 , S 中 1 0 P 效D会话 } 。

indication_cause

C 指 n D原因。 | G下 P 值之一:

AP_ACTIVATED

, S d 为活动D。

AP_DEACTIVATED

, S d 为 ; 活动D。

AP_PATH_SWITCHED

, S I 功 (" K通 7 * 关。

AP_SESS_COUNT_CHANGING

9 C D d , S D激活会话D } ? 。

AP_SETUP_FAILED

在完全激活 0 , S ' \ 。注意 RTP , S D统计信息只在 1 , S 转为 1 非活动状 , 1 5 现。**indication_cause** 为 AP_DEACTIVATED 或 AP_SETUP_FAILED。在 y P 其 { D i v 下 # t C r 。

RTP_INDICATION

rtp_stats.bytes_sent
C RTP ，S中, ＞X Z c已发M D总字Z ｝。

rtp_stats.bytes_received
C RTP ，S中, ＞X Z c已S U D总字Z ｝。

rtp_stats.bytes_resent
在转S中, ＞X Z c丢 ' 后重发D总字Z ｝。

rtp_stats.bytes_discarded
作为已U = D重４ ｝] 废弃D m一端 RTP ，S发M D (文总？。

rtp_stats.packets_sent
C RTP ，S中＞X Z c已发M D (文总？。

rtp_stats.packets_received
C RTP ，S中＞X Z c已U = D (文总？。

rtp_stats.packets_resent
转发１＞X Z c自己丢 ' 后重发D (文总？。

rtp_stats.packets_discarded
作为已U = D重４ ｝] 废弃D m一端 RTP ，S发M D (文总？。

rtp_stats.gaps_detected
＞X Z c检b = D间 t 总？。？一v间 t 代m着一v或多v丢 ' D帧。

rtp_stats.send_rate
C RTP ，S中１０D发M Y J （C ' 位？k b ？）。b G y] ARB c法计c D最大允许D发M Y J 。

rtp_stats.max_send_rate
C RTP ，S中最大D发M Y J （C ' 位？k b ？）。

rtp_stats.min_send_rate
C RTP ，S中最小D发M Y J （C ' 位？k b ？）。

rtp_stats.receive_rate
C RTP ，S中１０D S U Y J （C ' 位？k b ？）。b G通过最后b ？D１间间 t 计c v来D５际S U Y J 。

rtp_stats.max_receive_rate
C RTP ，S中最大D S U Y J （C ' 位？k b ？）。

rtp_stats.min_receive_rate
C RTP ，S中最小D S U Y J （C ' 位？k b ？）。

rtp_stats.burst_size
C RTP ，S中C字Z b ？D１０突发大小。

rtp_stats.up_time
C RTP ，S已活动D总D k ｝。

rtp_stats.smooth_rtt
在＞X Z c和对方 RTP Z c之间, 平滑b ？D回７１间（C毫k b ？）。

rtp_stats.last_rtt
＞X Z c和对方 RTP Z c之间最后b ？D回７１间（C毫k b ？）。

rtp_stats.short_req_timer

C Z SHORT_REQ 定 1 器 D 1 0 V续 1 间（C毫 k b ？)。

rtp_stats.short_req_timeouts

对 Z C RTP ，S SHORT_REQ 定 1 器已，1 总？。

rtp_stats.liveness_timeouts

对 Z C RTP ，S 运行着 D 定 1 器已，1 总？。1 ，S 在 **rtp_connection_detail.liveness_timer** 中指定 D 期间中 U 闲，运行着 D 定 1 器，1 。

rtp_stats.in_invalid_sna_frames

C RTP ，S 中作为无效 D S U 和废弃 D SNA 帧 D 总？。

rtp_stats.in_sc_frames

C RTP ，S 中 S U D 会话 X 制帧 D 总？。

rtp_stats.out_sc_frames

C RTP ，S 中发 M D 会话 X 制帧 D 总？。

SESSION_INDICATION

会话启动或M放 1 z I C指 n 。会话M放 1 ， + 会返回最后 D 会话统计信息。会话激活 1 ， # t **sess_stats** r 。

VCB Structure

```
typedef struct session_indication
{
   unsigned short  opcode;              /* verb operation code           */
   unsigned char   reserv2;             /* reserved                      */
   unsigned char   format;              /* format                        */
   unsigned short  primary_rc;          /* primary return code           */
   unsigned long   secondary_rc;        /* secondary return code         */
   unsigned char   data_lost;           /* previous indication lost      */
   unsigned char   deactivated;         /* has session been deactivated? */
   unsigned char   lu_name[8];          /* LU name                       */
   unsigned char   lu_alias[8];         /* LU alias                      */
   unsigned char   plu_alias[8];        /* partner LU alias              */
   unsigned char   fqplu_name[17];      /* fully qualified partner       */
                                        /* LU name                       */
   unsigned char   mode_name[8];        /* mode name                     */
   unsigned char   session_id[8];       /* session ID                    */
   FQPCID          fqpcid;              /* fully qualified procedure     */
   unsigned long   sense_data;          /* sense_data                    */
   unsigned char   duplex_support;      /* full-duplex support           */
   SESSION_STATS   sess_stats;          /* session statistics            */
   unsigned char   sscp_id[6];          /* SSCP ID of host               */
   unsigned char   plu_slu_comp_lvl;    /* PLU to SLU compression level  */
   unsigned char   slu_plu_comp_lvl;    /* SLU to PLU compressionlevel   */
                                        /* correlator ID                 */
   unsigned char   reserva[12];         /* reserved                      */
} SESSION_INDICATION;

typedef struct fqpcid
{
   unsigned char   pcid[8];             /* procedure correlator          */
                                        /* identifier                    */
   unsigned char   fqcp_name[17];       /* originator's network          */
                                        /* qualified CP name             */
   unsigned char   reserve3[3];         /* reserved                      */
} FQPCID;

typedef struct session_stats
{
   unsigned short  rcv_ru_size;         /* session receive RU size       */
   unsigned short  send_ru_size;        /* session send RU size          */
   unsigned short  max_send_btu_size;   /* max send BTU  size            */
   unsigned short  max_rcv_btu_size;    /* max rcv BTU size              */
   unsigned short  max_send_pac_win;    /* max send pacing window size   */
   unsigned short  cur_send_pac_win;    /* curr send pacing window size  */
   unsigned short  max_rcv_pac_win;     /* max receive pacing win size   */
   unsigned short  cur_rcv_pac_win;     /* curr receive pacing win size  */
   unsigned long   send_data_frames;    /* number of data frames sent    */
   unsigned long   send_fmd_data_frames;
                                        /* num FMD data frames sent      */
   unsigned long   send_data_bytes;     /* number of data bytes sent     */
   unsigned long   rcv_data_frames;     /* num data frames received      */
   unsigned long   rcv_fmd_data_frames;
                                        /* num FMD data frames received  */
   unsigned long   rcv_data_bytes;      /* num data bytes received       */
   unsigned char   sidh;                /* session ID high byte          */
   unsigned char   sidl;                /* session ID low byte           */
   unsigned char   odai;                /* ODAI bit set                  */
```

```
   unsigned char   ls_name[8];          /* Link station name            */
   unsigned char   pacing_type;         /* type of pacing in use        */
   unsigned char   reserve;             /* reserved                     */
} SESSION_STATS;
```

Parameters

opcode

AP_SESSION_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

primary_rc

AP_OK

secondary_rc

H Z c 。

data_lost

5 w }] G否已 - 丢 '（AP_YES 或 AP_NO）。 1 内 ? 组件检 b = 一次引起 0 一 u 指 n 丢 ' D故O后 h 置。如果 **data_lost** j 志位置为 AP_YES，那 4 f 后 D }] r 能置为U。 C & C L 序 & 发 v 一 u QUERY | n C以 | 新已丢 ' D信息。

deactivated

1 会话启动 1 置为 AP_NO。 1 会话M放 1 置为 AP_YES。

lu_name

LU { F 。 C { F G v 8 字Z D A 型 EBCDIC 字符串。

lu_alias

> X定义D LU p { 。 b G v以 I 显 > D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。

plu_alias

对方 LU D p { 。 b G v以 I 显 > D > X字符集m > D 8 字Z D字符串。

fqplu_name

C Z伙 i LU D 17 字Z完 { 7 6网 g { 。 C { F G I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构 I， " C EBCDIC U q R n d。（ ? v { F D $ 度最多 I 为 8 v字Z， R ；含 6 入U q 。）

mode_name

为一组会话指定网 g t 性D方 = { 。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d。

session_id

会话D 8 v字Z j 6 符。

fqpcid.pcid

Procedure correlator ID。 C ID 为 8 字Z D . y x 制字符串。

fqpcid.fqcp_name

完 { 7 6 X制 c D { 字。 { F $ 度为 17 v字Z， " 以 EBCDIC U q R n d。 | I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c ，S。（ ? v { F D $ 度最多 I 为 8 v字Z， R ；含 6 入U q 。）

SESSION_INDICATION

sense_data

在b } k s 1发M或S U D检b }]。如 **deactivated** 为 AP_NO，则# t C r。

duplex_support

如同在，S中b v D，返回通话+工支V。 | G下P值之一:

AP_HALF_DUPLEX

只支V k +工通话。

AP_FULL_DUPLEX

全+工和k +工通话一样支V。

AP_UNKNOWN

因为；P同对方 LU 激活D会话，通话D +工支V；能x行。

sess_stats.rcv_ru_size

最大S U RU 大小。

sess_stats.send_ru_size

最大发M RU 大小。

sess_stats.max_send_btu_size

能发M D最大 BTU 大小。

sess_stats.max_rcv_btu_size

能S U D最大 BTU 大小。

sess_stats.max_send_pac_win

会话中发M w =窗Z D最大_寸。

sess_stats.cur_send_pac_win

会话中发M w =窗Z D 1 0大小。

sess_stats.max_rcv_pac_win

会话中S U w =窗Z D最大_寸。

sess_stats.cur_rcv_pac_win

会话中S U w =窗Z D 1 0大小。

sess_stats.send_data_frames

发M D } # w }]帧D } ?。

sess_stats.send_fmd_data_frames

发M D } # w FMD }]帧D } ?。

sess_stats.send_data_bytes

发M D } # }] w D字Z }。

sess_stats.rcv_data_frames

S U D } # w }]帧D } ?。

sess_stats.rcv_fmd_data_frames

S U D } # w FMD }]帧D } ?。

sess_stats.rcv_data_bytes

S U D } # }] w D字Z }。

sess_stats.sidh

Session ID _字Z。

sess_stats.sidl

Session ID M字Z。

sess_stats.odai

原<目DX址指>符。1("会话1，如果>XZc|含主47>，BIND D发M方置Cr为0; "R如果 BIND D发M方G|含从47>DZc，则置Cr为1。

sess_stats_stats.ls_name

同统计信息相关*D47>{F。bGv以I显>D>X字符集m>D 8字ZD字符串。yPD 8 v字ZGP效D。Cr能CZQ会话D统计信息通过信息w?同47关*起来。

sess_stats_pacing_type

会话中9CDSU同=类型。CN}能取 AP_NONE、AP_PACING_ADAPTVE 或 AP_PACING_FIXED H值。

sscp_id

对Z关* LU 会话，C字段|含着从CZ>X LU 3d PU D主机D ACTPU 中SUD SSCP ID。对Z独" LU 会话，Cr+置为全?二x制c。

plu_slu_comp_lvl

指定从 PLU 发M= SLU D}]D压uL度。

AP_NONE

；9C压u。

AP_RLE_COMPRESSION

9C RLE 压u。

AP_LZ9_COMPRESSION

CZc能支V LZ9 压u。

AP_LZ10_COMPRESSION

CZc能支V LZ10 压u。

AP_LZ12_COMPRESSION

CZc能支V LZ12 压u。

slu_plu_comp_lvl

指定从 SLU 发M= PLU D}]D压uL度。

AP_NONE

；9C压u。

AP_RLE_COMPRESSION

9C RLE 压u。

AP_LZ9_COMPRESSION

CZc能支V LZ9 压u。

AP_LZ10_COMPRESSION

CZc能支V LZ10 压u。

AP_LZ12_COMPRESSION

CZc能支V LZ12 压u。

SESSION_FAILURE_INDICATION

无 [何 1 会话M放，都会 z I C指 n 。 # 证K C指 n ； 即， I 功 z I C指 n 。

VCB Structure

```
typedef struct session_failure_indication
{
   unsigned short  opcode;              /* verb operation code          */
   unsigned char   reserv2;             /* reserved                     */
   unsigned char   format;              /* format                       */
   unsigned short  primary_rc;          /* primary return code          */
   unsigned long   secondary_rc;        /* secondary return code        */
   unsigned char   reserv3[3];          /* reserved                     */
   unsigned char   lu_name[8];          /* LU name                      */
   unsigned char   lu_alias[8];         /* LU alias                     */
   unsigned char   plu_alias[8];        /* partner LU alias             */
   unsigned char   fqplu_name[17];      /* fully qualified partner      */
                                        /* LU name                      */
   unsigned char   mode_name[8];        /* mode name                    */
   unsigned char   session_id[8];       /* session ID                   */
   unsigned long   sense_data;          /* sense_data                   */
  } SESSION_FAILURE_INDICATION;
```

Parameters

opcode

AP_SESSION_FAILURE_INDICATION

format

j 6 VCB D q = 。 + C字段 h 置为 c ，以指定O f P v D VCB D f > 。

primary_rc

AP_OK

lu_name

LU { F 。 C { F G v 8 字Z D A 型 EBCDIC 字符串。

lu_alias

> X定义D LU p { 。 b G v以 I 显 > D > X字符集 m > D 8 字Z D字符串。 y P D 8 v字Z G P效D。

plu_alias

对方 LU D p { 。 b G v以 I 显 > D > X字符集 m > D 8 字Z D字符串。

fqplu_name

C Z伙 i LU D 17 字Z完 { {

UNREGISTER_INDICATION_SINK

UNREGISTER_INDICATION_SINK > } 那些 S U 未 k s 指 n D x L 和队 P D j 6。

如果指定 D 组合 **proc_id**、**queue_id**、 和 **indication_opcode** 曾 - 注 a 过，则 > } C j 6。如果指定 D 组合曾 - 注 a 过多次，则 > } k UNREGISTER_INDICATION_SINK D 信息头 verb_signal 中 D **orig_verb_data** 相匹配 D j 6。

VCB Structure

```
typedef struct port_indication
{
   unsigned short  opcode;               /* verb operation code          */
   unsigned char   reserv2;              /* reserved                     */
   unsigned char   format;               /* format                       */
   unsigned short  primary_rc;           /* primary return code          */
   unsigned long   secondary_rc;         /* secondary return code        */
   unsigned PROC_ID
                   proc_id;              /* process identifier of sink   */
   unsigned QUEUE_ID
                   queue_id;             /* queue identifier where       */
                                         /* indications will be sent     */
   unsigned short  indication_opcode;    /* opcode of indication to      */
                                         /* be sunk                      */
} REGISTER_INDICATION_SINK;
```

Parameters

opcode

AP_UNREGISTER_INDICATION_SINK

format

j 6 VCB D q = 。 + C 字段 h 置为 c，以指定 O f P v D VCB D f > 。

proc_id

发 M 指 n D Process ID D 处理。

queue_id

发 M 指 n D Queue ID D 队 P 。

indication_opcode

返回指 n D Opcode。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N }:

primary_rc

AP_OK

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_OP_CODE

AP_DYNAMIC_LOAD_ALREADY_REGD

如果C动词因状，错误而未能执行，则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_LU_NAME

如果C动词因}确D START_NODE N } ; P发M而未能执行，则L序返回下P N }:

primary_rc

AP_FUNCTION_NOT_SUPPORTED

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因 STOP_NODE | n P未发v而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

第10章 2全T动词

>BZhvKCZ定义和>}2全性ZnD动词。

© Copyright IBM Corp. 1989, 1997, 1998

CONV_SECURITY_BYPASS

CONV_SECURITY_BYPASS 允许& C X制L 序G否加？ > X LU D对话级 p 2 全性。一) \ * K 2 全性，则L 序 + ；会对 > X LU O D对话做任何认证或 Z 权。

VCB a 构

```
typedef struct conv_security_bypass
{
   unsigned short  opcode;           /* verb operation code        */
   unsigned char   reserv2;          /* reserved                   */
   unsigned char   format;           /* format                     */
   unsigned short  primary_rc;       /* primary return code        */
   unsigned long   secondary_rc;     /* secondary return code      */
   unsigned char   lu_name[8];       /* local LU name              */
   unsigned char   lu_alias[8];      /* local LU alias             */
   unsigned char   bypass_security;  /* should security be         */
                                     /* bypassed?                  */
   unsigned char   reserv3[3];       /* reserved                   */
} DEFINE_LU_LU_PASSWORD;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_CONV_SECURITY_BYPASS

format

j 6 VCB D q =。 + C字段 h 置为 c，以指定O f P v D VCB D f >。

lu_name

> X LU D LU { 字。 C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C 字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU 。

lu_alias

> X LU D p { 。 b G v 以 I 显 > D > X字符集m > D 8 字Z D字符串。 v 1 **lu_name** 字段 h 置为全 c 1， | E P 效，在 b 种 i v 下。 y P 8 v字Z 都G P 效D " R X须 h 置。如果 **lu_alias** 和 **lu_name** = _ 置为全 c，则C 动词转发 x 同X制 c 相关 * D LU（缺！ LU)。

bypass_security

指定G 否 \ * 2 全性（AP_YES 或 AP_NO)。

返回N数

如果C 动词 I 功执行，则L 序返回下 P N }:

primary_rc

AP_OK

如果C 动词因N } 错误而未能执行，则L 序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

AP_INVALID_LU_ALIAS
AP_INVALID_BYPASS_SECURITY

如果C动词因ZcP未启动而未能执行，则L序返回下PN}:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因Zc停止而未能执行，则L序返回下PN}:

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

CREATE_PASSWORD_SUBSTITUTE

CREATE_PASSWORD_SUBSTITUTE 返回C Z z I f 换和指定会话D检 i 器D Z n f 换、Z n检 i 器和发M D序P号。

VCB a 构

```
typedef struct create_password_substitute
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   lu_alias[8];       /* LU alias                     */
   unsigned char   conv_group_id[8];  /* partner LU alias             */
   unsigned char   user_id[10];       /* user ID                      */
   unsigned char   pw[10];            /* clear text password          */
   unsigned char   seq_no[8];         /* sequence number              */
   unsigned char   pw_sub[10];        /* password substitute          */
   unsigned char   pw_verifier[10];   /* password verifier            */
} CREATE_PASSWORD_SUBSTITUTE;
```

提供N数

&C L序a供下P N }:

opcode

AP_CREATE_PASSWORD_SUBSTITUTE

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

lu_alias

> X定义D LU p {。b G v以I显> D > X字符集m > D 8 字Z D字符串。

conv_group_id

I LU 9 C D会话对话组j 6符。

user_id

C户 ID 。

pw 在加\ c法中9 C D w k通信 (文。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

seq_no

发M在加\ c法中9 C D 3序号。注意，如果C动词I功执行，则b次会话发M 3序号D内？值+增加。返回D C值G增加后D值。

pw_sub

I加\ c法z I D Z n f 换。

pw_verifier
I 加 \ c 法 z I D Z n 检 i 器。

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_BAD_LU_ALIAS

AP_DEACT_CG_INVALID_CGID

如果 C 动词因 Z c ；支 V Z n f 换而未能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_PW_SUB_NOT_SUPP_ON_SESS

如果 C 动词因 Z c P 未启动而未能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果 C 动词因系统错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_LU_LU_PASSWORD

DEFINE_LU_LU_PASSWORD a 供K C Z > X LU 和伙 i LU 间会话级验证D Z n。

VCB a 构

```
typedef struct define_lu_lu_password
{
   unsigned short  opcode;           /* verb operation code        */
   unsigned char   reserv2;          /* reserved                   */
   unsigned char   format;           /* format                     */
   unsigned short  primary_rc;       /* primary return code        */
   unsigned long   secondary_rc;     /* secondary return code      */
   unsigned char   lu_name[8];       /* local LU name              */
   unsigned char   lu_alias[8];      /* local LU alias             */
   unsigned char   fqplu_name[17];   /* fully qualified partner    */
                                     /* LU name                    */
   unsigned char   verification_protocol
                                     /* LULU verification protocol */
   unsigned char   description[RD_LEN];
                                     /* resource description       */
   unsigned char   reserv3[8];       /* reserved                   */
   unsigned char   password[8];      /* password                   */
} DEFINE_LU_LU_PASSWORD;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_DEFINE_LU_LU_PASSWORD

format

j 6 VCB D q =。+ C字段 h 置为 c，以指定O f P v D VCB D f >。

lu_name

> X LU D LU {字。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU。

lu_alias

> X LU D p {。b G v 以 I 显 > D > X字符集m > D 8 字Z D字符串。在 y P D 8 v字Z P效R X须 h 置D i v下，如果 **lu_name** 字段置为全 c，则C字段 v 在此 1 P 效。如果 **lu_alias** 和 **lu_name** = _ 置为全 c，则C动词转发 x 同X制 c 相关 * D LU（缺！ LU)。

fqplu_name

完{ 7 6 伙 i LU D { 字。{ 字为 17 字Z $ " R I EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。(？v { F D $ 度最多 I 为 8 v字Z，R；含 6 入U q。)

verification_protocol

C Z k 伙 i LU 一起 9 C D LU-LU 验证协议:

AP_BASIC_PROTOCOL

v P 基 > 协议 + k 伙 i LU 一起 9 C。

AP_ENHANCED_PROTOCOL

v P 增？协议 + k 伙 i LU 一起 9 C。

AP_EITHER_PROTOCOL

要 4 G基 > 协议，要 4 G增？协议能够 k C伙 i LU 一起 9 C，转向下 P 详细 i v D主 b：

- C字段缺！D h 置G AP_EITHER_PROTOCOL。
- AP_EITHER_PROTOCOL b v 值 a 供K对移植 = 增？协议 9 C D推动。直 = 伙 i LU 一) 许 I 运行增？协议，> X LU E S \ 基 > 协议。从那之后，} 非发 v DEFINE_LU_LU_PASSWORD 以允许基 > 协议，否则 + ； S \ | 。

description

资源 5 w。

password

Z n。 b G v 8 字 Z D . y x 制字符串。注意 Z n 中D ？ v 字 Z 最小 P 效位；在会话级验证中 9 C。

返回N数

如果C动词 I 功执行，则L序返回下 P N }：

primary_rc

AP_OK

如果C动词因N } 错误而未能执行，则L序返回下 P N }：

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PLU_NAME

AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS

如果C动词因 Z c P 未启动而未能执行，则L序返回下 P N }：

primary_rc

AP_NODE_NOT_STARTED

如果C动词因 Z c 停止而未能执行，则L序返回下 P N }：

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下 P N }：

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DEFINE_USERID_PASSWORD

DEFINE_USERID_PASSWORD 定义K k C户 ID 相关 * D Z n。

VCB a构

```
define_userid_password
{
   unsigned short  opcode;              /* verb operation code        */
   unsigned char   reserv2;             /* reserved                   */
   unsigned char   format;              /* format                     */
   unsigned short  primary_rc;          /* primary return code        */
   unsigned long   secondary_rc;        /* secondary return code      */
   unsigned short  define_type;         /* what the define type is    */
   unsigned char   user_id[10];         /* user id                    */
   unsigned char   reserv3[8];          /* reserved                   */
   USERID_PASSWORD_CHARS password_chars;
                                        /* password characteristics   */
} DEFINE_USERID_PASSWORD;

typedef struct userid_password_chars
{
   unsigned char   description[RD_LEN];
                                        /* resource description       */
   unsigned short  profile_count;       /* number of profiles         */
   unsigned short  reserv1;             /* reserved                   */
   unsigned char   password[10];        /* password                   */
   unsigned char   profiles[10][10];    /* profiles                   */
} USERID_PASSWORD_CHARS;
```

提供N数

& C L序a供下P N }:

opcode

AP_DEFINE_USERID_PASSWORD

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

define_type

指定C户Z n定义D类型:

AP_ADD_USER

指定新C户，或 _ 为现P C户D d Z n。

AP_ADD_PROFILES

为现P C户D启动文件指定m加。

user_id

C户j 6符。 b G v 10 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。

password_chars.description

资源5 w。 b G v以I显 > D > X字符集m > D 16 字Z D字符串。 y P D 16 v字Z G P效D。

password_chars.profile_count

启动文件D } 目。

password_chars.password
C户D Z n。 b G v 10 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d。

password_chars.profiles
k C户相关 * D启动文件。 b些中D ？一 v G 10 字Z D AE 型 EBCDIC 字符串， I EBCDIC U q R n d。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc
AP_OK

如果C动词因N } 错误而未能执行，则L序返回下P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_NO_PROFILES

AP_UNKNOWN_USER
AP_INVALID_UPDATE_TYPE
AP_TOO_MANY_PROFILES
AP_INVALID_USERID
AP_INVALID_PROFILE
AP_INVALID_PASSWORD

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c 停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

DELETE_LU_LU_PASSWORD

DELETE_LU_LU_PASSWORD > } LU-LU Z n。

VCB a 构

```
typedef struct delete_lu_lu_password
{
   unsigned short  opcode;           /* verb operation code        */
   unsigned char   reserv2;          /* reserved                   */
   unsigned char   format;           /* format                     */
   unsigned short  primary_rc;       /* primary return code        */
   unsigned long   secondary_rc;     /* secondary return code      */
   unsigned char   lu_name[8];       /* LU name                    */
   unsigned char   lu_alias[8];      /* local LU alias             */
   unsigned char   fqplu_name[17];   /* fully qualified partner    */
                                     /* LU name                    */
   unsigned char   reserv3;          /* reserved                   */
} DELETE_LU_LU_PASSWORD;
```

提供N数

& C L 序 a 供下 P N }:

opcode

AP_DELETE_LU_LU_PASSWORD

format

j 6 VCB D q =。+ C字段 h 置为 c，以指定O f P v D VCB D f >。

lu_name

> X LU D LU { 字。C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段置为全 c，则 **lu_alias** 字段 + C Z v 定 > X LU。

lu_alias

> X LU D p { 。b G v 以 I 显 > D > X字符集 m > D 8 字Z D字符串。v 1 **lu_name** 字段 h 置为全 c 1，| E P 效，在 b 种 i v 下。y P 8 v字Z都G P 效D " R X须 h 置。如果 **lu_alias** 和 **lu_name** = _ 置为全 c，则C动词转发 x 同X制 c 相关 * D LU（缺！ LU)。

fqplu_name

完{ 7 6伙 i LU D { 字。{ F $ 度为 17 v字Z，"以 EBCDIC U q R n d。| I = v A 型 EBCDIC 字符串组 I，中间以一 v EBCDIC c，S。(? v { F D $ 度最多 I 为 8 v字Z，R；含 6 人U q。）

返回N数

如果C动词 I 功执行，则L序返回下 P N }:

primary_rc

AP_OK

如果C动词因N } 错误而未能执行，则L序返回下 P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PLU_NAME

AP_INVALID_LU_NAME
AP_INVALID_LU_ALIAS

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_USERID_PASSWORD

DELETE_USERID_PASSWORD > } k C户 ID 相关 * D Z n。

VCB a构

```
typedef struct delete_userid_password
{
   unsigned short  opcode;               /* verb operation code        */
   unsigned char   reserv2;              /* reserved                   */
   unsigned char   format;               /* format                     */
   unsigned short  primary_rc;           /* primary return code        */
   unsigned long   secondary_rc;         /* secondary return code      */
   unsigned short  delete_type;          /* type of delete             */
   unsigned char   user_id[10];          /* user id                    */
   USERID_PASSWORD_CHARS password_chars;
                                         /* password characteristics   */

} DELETE_USERID_PASSWORD;

typedef struct userid_password_chars
{
   unsigned char   description[RD_LEN];  /* resource description       */
   unsigned short  profile_count;        /* number of profiles         */
   unsigned short  reserv1;              /* reserved                   */
   unsigned char   password[10];         /* password                   */
   unsigned char   profiles[10][10];     /* profiles                   */
} USERID_PASSWORD_CHARS;
```

提供N数

& C L序 a 供下 P N }:

opcode

AP_DELETE_USERID_PASSWORD

format

j 6 VCB D q =。 + C字段 h 置为 c，以指定O f P v D VCB D f >。

delete_type

指定 > } D类型:

AP_REMOVE_USER

> } C户 Z n，和 y P 关 * D启动文件。

AP_REMOVE_PROFILES

> } 指定D启动文件。

user_id

C户 j 6符。 b G v 10 字 Z D字母 } 字集D A 型 EBCDIC 字符串（以一 v 字母 * < ）， I EBCDIC U q R n d。

password_chars.description

1 处理C 动词 1 忽 T C字段。

password_chars.profile_count

启动文件D } 目。

password_chars.password

1 处理C 动词 1 忽 T C字段。

password_chars.profiles

k C户相关 * D启动文件。 b些中D ？一 v G 10 字Z D AE 型 EBCDIC 字符串， I EBCDIC U q R n d。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果C动词因N }错误而未能执行，则L序返回下P N }:

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_NO_PROFILES

AP_UNKNOWN_USER
AP_INVALID_UPDATE_TYPE

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

SIGN_OFF

SIGN_OFF 指> LU 从G < P m中> } 项。 1 0， v > } 来自G < P m D项。 C动词能够指定G否> } y P项，或v > } m加D sign_off_data a 构中D那些项。

VCB a 构

```
typedef struct query_sign_off
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   lu_name[8];        /* LU name                      */
   unsigned char   lu_alias[8];       /* LU alias                     */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   fqplu_name[17];    /* fully qualified partner      */
                                      /* LU name                      */
   unsigned char   list;              /* signed on to/from list       */
   unsigned char   all_in_list;       /* sign off all entries in list */
   unsigned char   immediate;         /* remove entries immediately   */
   unsigned char   num_entries;       /* number of entries            */
} QUERY_SIGN_OFF;

typedef struct sign_off_data
{
   unsigned char   user_id[10];       /* user ID                      */
   unsigned char   all_profiles;     /* all profiles for this user    */
   unsigned char   profile[10];      /* specific profile             */
} SIGN_OFF_DATA;
```

提供N数

& C L 序a 供下P N }:

opcode

AP_CREATE_PASSWORD_SUBSTITUTE

format

j 6 VCB D q =。 + C字段h 置为c，以指定O f P v D VCB D f >。

lu_name

LU D { 字。 C { F G v 8 字Z D A 型 EBCDIC 字符串。如果C字段h 置为全c，则 **lu_alias** 字段+ C Z 确定w引。

lu_alias

> X定义D LU p { 。 b G v以I 显> D > X字符集m > D 8 字Z D字符串。 v 1 **lu_name** 字段h 置为全c 1， | E P效，在b 种i v下。 y P 8 v字Z 都G P效D " R X须h 置。如果 **lu_name** 和 **lu_alias** = v字段置为全c，则 9 C k X制c 相关* D LU（缺！ LU)。

plu_alias

伙i LU D p { 。 b G v以I 显> D > X字符集m > D 8 字Z D字符串。 y P 8 v字Z 都G P效D， " R X须h 置。如果C字段置为全c，则 **fqplu_name** 字段+ C Z v定w引值。

fqplu_name

伙i LU D 17 字Z完{ 7 6网g {。 b v { 字G I = v C EBCDIC c ， S

D A 型 EBCDIC 字符串构 I， R I EBCDIC U q R n d。（ ? v { F D $ 度最多 I 为 8 v 字 Z， R ；含 6 入 U q。）

list G < P m类型。 X须置为 AP_SIGNED_ON_TO_LIST 。

AP_SIGNED_ON_TO_LIST

从 > X LU G < = 远 L LU D C 户 P m。注意， 1 项从 P m

任何；P I L 序 I 功处理 D SIGN_OFF_DATA **user_id/profile** 组合， + 返回 x m 加 = VCB D & C L 序， " R I L 序返回 D **num_entries** 返回值 G SIGN_OFF_DATA 项 D 号 k （ ；能处理)。

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LU_ALIAS

AP_INVALID_LU_NAME
AP_INVALID_LU_NAME
AP_INVALID_LIST

如果 C 动词因 Z c P 未启动而未能执行，则 L 序返回下 P N } :

primary_rc
AP_NODE_NOT_STARTED

如果 C 动词因 Z c 停止而未能执行，则 L 序返回下 P N } :

primary_rc
AP_NODE_STOPPING

如果 C 动词因系统错误而未能执行，则 L 序返回下 P N } :

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

第11章 APING 和 CPI-C 动词

> B Z h v K C Z 『ping』 其 | Z c 和 C Z 定义、> } 和 i 询 CPI-C 9 d 信息 D 动词。

© Copyright IBM Corp. 1989, 1997, 1998

APING

APING 允许管理& C L 序去 『ping』 在网 g 中D远L LU 。 1 **partner_ver_len** 字段置为大Z c D值 1， 验证 }] 串（指定D $ 度）能够m加 = VCB D末端 " R返回。

v 人通信或通信服务器 APING 作为内 ? 9 C v 人通信或通信服务器 APPC API D 『服务B务L序5现』（在 *v 人通信M户机/服务器通信L序h计* 中h v D)。

VCB a 构

```
typedef struct aping
{
   unsigned short  opcode;            /* verb operation code          */
   unsigned char   reserv2;           /* reserved                     */
   unsigned char   format;            /* format                       */
   unsigned short  primary_rc;        /* primary return code          */
   unsigned long   secondary_rc;      /* secondary return code        */
   unsigned char   lu_name[8];        /* local LU name                */
   unsigned char   lu_alias[8];       /* local LU alias               */
   unsigned long   sense_data;        /* sense data                   */
   unsigned char   plu_alias[8];      /* partner LU alias             */
   unsigned char   mode_name[8];      /* mode name                    */
   unsigned char   tp_name[64];       /* destination TP name          */
   unsigned char   security;          /* security level               */
   unsigned char   reserv3a[3];       /* reserved                     */
   unsigned char   pwd[10];           /* password                     */
   unsigned char   user_id[10];       /* user ID                      */
   unsigned short  dlen;              /* length of data to send       */
   unsigned short  consec;            /* number of consecutive sends  */
   unsigned char   fqplu_name[17];    /* fully qualified partner      */
                                      /* LU name                      */
   unsigned char   echo;              /* data echo flag               */
   unsigned short  iterations;        /* number of iterations         */
   unsigned long   alloc_time;        /* time taken for ALLOCATE      */
   unsigned long   min_time;          /* min send/receive time        */
   unsigned long   avg_time;          /* average send/receive time    */
   unsigned long   max_time;          /* max send/receive time        */
   unsigned short  partner_ver_len;   /* size of string to receive    */
} APING;
```

提供N数

& C L 序a 供下P N }:

opcode

AP_APING

format

j 6 VCB D q = 。 + C字段h 置为c， 以指定O f P v D VCB D f > 。

lu_name

发v APING 动词D > X LU D LU D { 字。 C { F G v 8 字Z D A 型EBCDIC 字符串。 如果C字段置为全c， 则 **lu_alias** 字段 + C Z v 定 > X LU。

lu_alias

发M APING 动词D > X LU D p { 。 b G v 以I 显 > D > X字符集m > D 8字Z D字符串。 v 1 **lu_name** 字段h 置为全c 1， | E P效， 在b 种i v 下。

y P 8 v字Z都G P效D " R X须h置。如果 **lu_name** 和 **lu_alias** = _ 置为二x制c，则9 C缺！（X制c）LU 。

plu_alias

为> X B务L序I p { y知D伙i LU 。 b G v以I显> D > X字符集m > D 8 字Z D字符串。 y P D 8 v字Z G P效D " R X须h置。 b v { 字X须k在配置过L中 (" D伙i LU D { 字匹配。如果C N } 置为二x制c，则9 C **fqplu_name** N } 代f 。

mode_name

9 C D模= { 。 b G v 8 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。

tp_name

k s B务L序D { 字。 b G v 64 位字符串。 Z c Y作员；检i C字符串D字符集。**tp_name** D值X须k在远L LU O配置D相匹配。在 EBCDIC k中， C字符串通 # 置为I EBCDIC U q R n d D 『APINGD』 。

security

为K验证对k s B务L序D访问，要指定伙i LU y X需D信息:

AP_NONE
AP_PGM
AP_SAME
AP_PGM_STRONG

pwd 同 **user_id** 相关 * D Z n。 b G v 10 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。 v 1 如果 **security** 置为 AP_PGM 或 AP_PGM_STRONG 1需要。

user_id

访问伙i B务L序y X需D C户 ID 。 b G v 10 字Z D字母 } 字集D A 型 EBCDIC 字符串（以一v字母 * < ）， I EBCDIC U q R n d。如果 **security** 置为 AP_PGM、AP_PGM_STRONG 或 AP_SAME 1需要。

dlen I APING B务L序发M D }] $ 度。APING 发M $ 度为 **dlen** D c字符串。

consec

在？次 | 代1 ，续发M执行D号k。APING 发v ？一v I **dlen** 字Z组I D }]，MC_SEND_DATA 动词D b v号k。如果 **echo** N } 置为 AP_YES，则 APING j记最后D MC_SEND_DATA 为 AP_SEND_DATA_P_TO_R_FLUSH（准8 S U e U）" R H待来自伙i APINGD B务L序 | 含 }] D响&（通过发v MC_RECEIVE_AND_WAIT）。如果 **echo** N } 置为 AP_NO，则 APING e U }] " R H待确认（通过j记 MC_SEND_DATA 为 AP_SEND_DATA_CONFIRM）。在= 种i v下， 3序h v K b里同 SNA 4相一致D X方。

fqplu_name

伙i LU D 17 字Z完{ 7 6网g { F。 b v { 字G I = v C EBCDIC c串 * D A 型 EBCDIC 字符串构I， R I EBCDIC U q R n d。（？v { F D $ 度最多I为 8 v字Z， R ；含6入U q。）v 1 **plu_alias** 字段h置为全c 1， | E P效。

echo 指定1 APING B务L序a x发M y需要 } ？D }] 1， | G否期待响&:

AP_YES
AP_NO

iterations
I APING 发 v D，续序 P D | 代号 k（ I **consec** N } 定义 D）。在 SNA u o 中， b v N } 定义 K + 要发 M D 4 D 号 k。

partner_ver_len
伙 i B 务 L 序验证 }] 串能够 I 管理 L 序 S U D 最大 $ 度。

返回 N 数

如果动词 I 功执行，则 APING 返回下 P N }：

primary_rc
AP_OK

sense_data
如果动词 I 功返回， C 字段 + 为 c。

alloc_time
MC_ALLOCATE = 远 L B 务 L 序 a x y 需 D 1 间（毫 k 级）。

min_time
}] 发 M | 代 y 需最小 1 间（毫 k 级）。 C N } | 括伙 i 响 & y 需 D 1 间。（要 4 通过发 M }]，要 4 通过发 v 确认，取 v Z **echo** N } D h 置）。

avg_time
}] 发 M | 代 y 需 D 平 y 1 间（毫 k 级）。 C N } | 括伙 i 响 & y 需 D 1 间。（要 4 通过发 M }]，要 4 通过发 v 确认，取 v Z **echo** N } D h 置）。

max_time
}] 发 M | 代 y 需 D 最大 1 间（毫 k 级）。 C N } | 括伙 i 响 & y 需 D 1 间。（要 4 通过发 M }]，要 4 通过发 v 确认，取 v Z **echo** N } D h 置）。

partner_ver_len
伙 i B 务 L 序返回 D 验证字符串 D $ 度。字符串自 m m 加在 VCB D 末端。

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N }：

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_LU_NAME

AP_INVALID_LU_ALIAS

APING 9 C I v 人通信或通信服务器 APPC API a 供 D MC_ALLOCATE、MC_SEND_DATA、MC_RECEIVE_AND_WAIT、MC_CONFIRM 和 MC_DEALLOCATE 动词。在；I 功执行 D i v 下， b 些动词返回 D N } 在 *v 人通信 M 户机/服务器通信 L 序 h 计 中记载*。

如果 C 动词因 Z c P 未启动而未能执行，则 L 序返回下 P N }：

primary_rc
AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc
AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

CPI-C 动词

DEFINE_CPIC_SIDE_INFO

C动词增加或f换内存中D 9 d信息u目。CPI-C 9 d信息u目同带P符号目j { D一组对话X性相关*。如果在内存中已P作为C动词a供D带相同符号目j { D 9 d信息u目，则+ I b次w C a供D }]重写。k N阅 *CPI-C N* <大全 以获C | 多P关Z v人通信或通信服务器a供D CPI-C 支V D信息。

VCB a构

```
typedef struct define_cpic_side_info
{
   unsigned short  opcode;             /* verb operation code          */
   unsigned char   reserv2;            /* reserved                     */
   unsigned char   format;             /* format                       */
   unsigned short  primary_rc;         /* primary return code          */
   unsigned long   secondary_rc;       /* secondary return code        */
   unsigned char   reserv2a[8];        /* reserved                     */
   unsigned char   sym_dest_name[8];   /* Symbolic destination name    */
   CPIC_SIDE_INFO_DEF_DATA def_data;   /* defined data                 */
} DEFINE_CPIC_SIDE_INFO;

typedef struct cpic_side_info_def_data
{
   unsigned char   description[RD_LEN];
                                       /* resource description         */
   CPIC_SIDE_INFO  side_info;          /* CPIC side info               */
   unsigned char   user_data[32];      /* User defined data            */
} CPIC_SIDE_INFO_DEF_DATA;

typedef struct cpic_side_info
{
   unsigned char   partner_lu_name[17];
                                       /* Fully qualified partner      */
                                       /* LU name                      */
   unsigned char   reserved[3];        /* Reserved                     */
   unsigned long   tp_name_type;       /* TP name type                 */
   unsigned char   tp_name[64];        /* TP name                      */
   unsigned char   mode_name[8];       /* Mode name                    */
   unsigned long   conversation_security_type;
                                       /* Conversation security type   */
   unsigned char   security_user_id[CPIC_SECURITY_INFO_LEN];
                                       /* User ID                      */
   unsigned char   security_password[CPIC_SECURITY_INFO_LEN];
                                       /* Password                     */
} CPIC_SIDE_INFO;
```

提供N数

&C L序a供下P N }:

opcode

AP_DEFINE_CPIC_SIDE_INFO

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

sym_dest_name

j 6 9 d信息u目D符号目j {。b v { 字以I显> D字符集m >，最大 8 字Z $， I U q R n d。C允许D字符G大写字母（A = Z）和}字 0 = 9。

def_data.description

资源5 w（在 QUERY_CPIC_SIDE_INFO 中返回）。b G v在I显> D > X字符集中D 16 字Z D字符串。y P D 16 v字Z G P效D。

def_data.side_info.partner_lu_name

伙 i LU D完 { 7 6 { 。 b v { 字为 17 字 Z $ R I EBCDIC U q R n d 。 | I = v C c ， S D字符串组 I 。（ ？ v { F D $ 度最多 I 为 8 v字 Z， R ；含 6 入 U q 。）

def_data.side_info.tp_name_type

B 务 L 序 { 字类型。 C字段置为下 P 之一:

XC_APPLICATION_TP

指定 y a 供 D B 务 L 序 { ； G 服务 B 务 L 序。在 B 务 L 序 { 字中指定 D y P D字符 X 须 G 以 I 显 > D > X 字符集 m > D P 效字符。

XC_SNA_SERVICE_TP

指定 y a 供 D B 务 L 序 { G 服务 B 务 L 序。 } K W字符，在 B 务 L 序中指定 D y P D字符 X 须 G 以 I 显 > D > X 字符集 m > D P 效字符。 W字符 X 须 G 在范围 X'01' = X'3F'之间，； | 括 X'0E' 和 X'0F' D . y x 制 } 字。

def_data.side_info.tp_name

B 务 L 序 { ，在 I 显 > > X 字符集中 D 8 字 Z 字符串， I U q R n d 。

def_data.side_info.mode_name

方 = { ，在 I 显 > > X 字符集中 D 8 字 Z 字符串， I U q R n d 。

def_data.side_info.conversation_security_type

对话 2 全性类型。 C字段置为下 P 之一:

XC_SECURITY_NONE
XC_SECURITY_SAME
XC_SECURITY_PROGRAM
XC_SECURITY_PROGRAM_STRONG.

def_data.side_info.security_user_id

C户 ID 。 v 人通信或通信服务器 + 会 9 C C字段加 ? 对话级 2 全性。

def_data.side_info.security_password

Z n 。 v 人通信或通信服务器 + 会 9 C C字段加 ? 对话级 2 全性。

def_data.user_data

C户 }] 。 C }] 在 QUERY_CPIC_SIDE_INFO 中返回， + G ； P I v 人通信或通信服务器 9 C 或 b M。

返回 N 数

如果 C 动词 I 功执行，则 L 序返回下 P N } :

primary_rc

AP_OK

如果 C 动词因 N } 错误而未能执行，则 L 序返回下 P N } :

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_SYM_DEST_NAME

AP_INVALID_LENGTH

如果C动词因ZcP未启动而未能执行，则L序返回下PN}:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因Zc停止而未能执行，则L序返回下PN}:

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下PN}:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

DELETE_CPIC_SIDE_INFO

C动词 > } K CPI-C 9 d信息u目。 k N阅 *CPI-C N* < 大全 以获C | 多P关Z v人通信或通信服务器a供D CPI-C 支V D信息。

VCB a构

```
typedef struct delete_cpic_side_info
{
   unsigned short  opcode;             /* verb operation code          */
   unsigned char   reserv2;            /* reserved                     */
   unsigned char   format;             /* format                       */
   unsigned short  primary_rc;         /* primary return code          */
   unsigned long   secondary_rc;       /* secondary return code        */
   unsigned char   reserv2a[8];        /* reserved                     */
   unsigned char   sym_dest_name[8];   /* Symbolic destination name    */
} DELETE_CPIC_SIDE_INFO;
```

提供N数

& C L序a供下P N }:

opcode

AP_DELETE_CPIC_SIDE_INFO

format

j 6 VCB D q =。 + C字段h置为c，以指定O f P v D VCB D f >。

sym_dest_name

j 6 9 d信息u目D符号目j {。 b v { 字以I显 > D字符集m >，最大 8 字Z $， I U q R n d。 C允许D字符G大写字母（A = Z）和 } 字 0 = 9。

返回N数

如果C动词I功执行，则L序返回下P N }:

primary_rc

AP_OK

如果C动词因状，错误而未能执行，则L序返回下P N }:

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_SYM_DEST_NAME

如果C动词因Z c P未启动而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_NOT_STARTED

如果C动词因Z c停止而未能执行，则L序返回下P N }:

primary_rc

AP_NODE_STOPPING

如果C动词因系统错误而未能执行，则L序返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_CPIC_SIDE_INFO

C动词返回K x定D符号目 j { D 9 d信息。 C信息作为P m返回。为# t X定D 9 d信息u目或X定D大？ u目，则 **sym_dest_name** 字段&C h置。否则C字段&C置为全 c 。

VCB a构

```
typedef struct query_cpic_side_info
{
   unsigned short  opcode;              /* verb operation code        */
   unsigned char   reserv2;             /* reserved                   */
   unsigned char   format;              /* format                     */
   unsigned short  primary_rc;          /* primary return code        */
   unsigned long   secondary_rc;        /* secondary return code      */
   unsigned char   *buf_ptr;            /* pointer to buffer          */
   unsigned long   buf_size;            /* buffer size                */
   unsigned long   total_buf_size;      /* total buffer size required */
   unsigned short  num_entries;         /* number of entries          */
   unsigned short  total_num_entries;   /* total number of entries    */
   unsigned char   list_options;        /* listing options            */
   unsigned char   reserv3;             /* reserved                   */
   unsigned char   sym_dest_name[8];    /* Symbolic destination name  */
} QUERY_CPIC_SIDE_INFO;

typedef struct cpic_side_info_data
{
   unsigned short  overlay_size;        /* size of this entry         */
   unsigned char   sym_dest_name[8];    /* Symbolic destination name  */
   unsigned char   reserv1[2];          /* reserved                   */
   CPIC_SIDE_INFO_DEF_DATA  def_data;
} CPIC_SIDE_INFO_DATA;

typedef struct cpic_side_info
{
   unsigned char   partner_lu_name[17];
                                        /* Fully qualified partner    */
                                        /* LU name                    */
   unsigned char   reserved[3];         /* Reserved                   */
   unsigned long   tp_name_type;        /* TP name type               */
   unsigned char   tp_name[64];         /* TP name                    */
   unsigned char   mode_name[8];        /* Mode name                  */
   unsigned long   conversation_security_type;
                                        /* Conversation security type */
   unsigned char   security_user_id[CPIC_SECURITY_INFO_LEN];
                                        /* User ID                    */
   unsigned char   security_password[CPIC_SECURITY_INFO_LEN];
                                        /* Password                   */
} CPIC_SIDE_INFO;

typedef struct cpic_side_info_def_data
{
   unsigned char   description[RD_LEN];
                                        /* resource description       */
   CPIC_SIDE_INFO  side_info;           /* CPIC side info             */
   unsigned char   user_data[32];       /* User defined data          */
} CPIC_SIDE_INFO_DEF_DATA;
```

提供N数

&C L序a供下P N }:

opcode

AP_QUERY_CPIC_SIDE_INFO

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定 O f P v D VCB D f > 。

buf_ptr

指向能够写入 P m信息缓 e D指 k 。

buf_size

y a 供D缓 e x D大小。返回 D }] + ; ，过 b v 大小。

num_entries

返回项D最大 } ? 。项D } ? + ；会，过 b v 值。 c 值 m > 无限制。

list_options

m w在 P m信息中& C返回 2 4 。 **sym_dest_name** 指定K（ k N阅以下）m > C Z 指定返回 D 5 际信息启动 c D w引值。

AP_FIRST_IN_LIST

忽 T w引值 " R返回 P m从 P m中 D Z 一 v u 目 * < 。

AP_LIST_FROM_NEXT

在一次 u 目 I a 供D w引值指定之后，返回 P m从 P m中D下一 u 目 * < 。

AP_LIST_INCLUSIVE

返回 P m从 w引值 y 指定D那项 * < 。

sym_dest_name

j 6 9 d 信息 u 目D符号目 j { 。 b v { 字以 I 显 > D字符集 m >，最大 8 字 Z $， I U q R n d 。C允许D字符G大写字母（A = Z）和 } 字 0 = 9。

返回N数

如果C动词 I 功执行，则L序返回下 P N } :

primary_rc

AP_OK

buf_size

在缓 e x 中返回D信息 $ 度。

total_buf_size

返回值 m w返回 y P ; k s D P m信息 y 需要D缓 e x D大小。 I 以 _ Z **buf_size** 。

num_entries

5 际返回项D } ? 。

total_num_entries

已 - 返回项D总 } 。 I 以 _ Z **num_entries** 。

cpic_side_info_data.overlay_size

C项中D字 Z }，即 = 下一 v 返回项（如果 P D话）D偏移 ? 。

cpic_side_info_data.sym_dest_name

返回D 9 d 信息 u 目符号目 j { 。

cpic_side_info_data.def_data

在 DEFINE_CPIC_SIDE_INFO 动词中 a 供D定义 CPI-C 9 d 信息。

": 在 DEFINE_CPIC_SIDE_INFO 已 I v 人通信或通信服务器处理之后，CPIC w C I 能 D d 在 b v 动词中返回 D 9 d 信息。

如果 C 动词因状，错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
AP_INVALID_SYM_DEST_NAME

如果 C 动词因 Z c P 未启动而未能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_NOT_STARTED

如果 C 动词因 Z c 停止而未能执行，则 L 序返回下 P N }:

primary_rc
AP_NODE_STOPPING

如果 C 动词因系统错误而未能执行，则 L 序返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

第12章 ， S 管m器动词

v 人通信或通信服务器 ， S 管理器C 来管理 APPC 或 CPI-C L 序D 启动。 P 关 ， S 管理器功能D 5 w 在 *v 人通信M户机/服务器通信L 序h 计*中 a 供。

v 人通信或通信服务器Z c Y 作员h) 支V X 制 ， S 管理器D 三v 动词。 b 些动词对9 C v 人通信或通信服务器Z c Y 作员h) D 任何& C L 序都G I C D。

© Copyright IBM Corp. 1989, 1997, 1998

DISABLE_ATTACH_MANAGER

v人通信或通信服务器，S管理器在Z c启动后I缺！值启C。C户I以发v C动词来{ C y P动，加载。C动词4位K启动B务L序之0 ，S管理器检i D全V j志。

VCB a构

```
typedef struct disable_am
{
   unsigned short   opcode;              /* Verb operation code       */
   unsigned char    reserv2;             /* reserved                  */
   unsigned char    format;              /* format                    */
   unsigned short   primary_rc;          /* Primary return code       */
   unsigned long    secondary_rc;        /* Secondary return code     */
} DISABLE_AM;
```

提供N数

&C L序a供K下P N }:

opcode

AP_DISABLE_ATTACH_MGR

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

返回N数

如果动词I功执行，则，S管理器返回下P N }:

primary_rc

AP_OK

如果因未启动Z c而9动词；能执行，则，S管理器返回下P N }:

primary_rc

AP_NODE_NOT STARTED

如果因系统错误而9动词；能执行，则，S管理器返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

ENABLE_ATTACH_MANAGER

如果，S管理器已{C，则I以通过发v v人通信或通信服务器Z c Y作员h)动词ENABLE_AM 来重新启C。 b h置K启动B务L序之0 ，S管理器检i D全V j志。

VCB a构

```
typedef struct enable_am
{
   unsigned short  opcode;           /* Verb operation code        */
   unsigned char   reserv2;          /* reserved                   */
   unsigned char   format;           /* format                     */
   unsigned short  primary_rc;       /* Primary return code        */
   unsigned long   secondary_rc;     /* Secondary return code      */
} ENABLE_AM
```

提供N数

&C L序a供K下P N }:

opcode

AP_ENABLE_ATTACH_MGR

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

返回N数

如果动词I功执行，则，S管理器返回下P N }:

primary_rc

AP_OK

如果因未启动Z c而9动词；能执行，则，S管理器返回下P N }:

primary_rc

AP_NODE_NOT STARTED

如果因系统错误而9动词；能执行，则，S管理器返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

QUERY_ATTACH_MANAGER

QUERY_ATTACH_MANAGER 动词 I C来发现，S管理器组件D状，，｜I以9 C ENABLE_ATTACH_MANAGER 和 DISABLE_ATTACH_MANAGER ｜n来启动和停止。

VCB a构

```
typedef struct query_am
{
        unsigned short   opcode;          /* Verb operation code          */
        unsigned char    reserv2;         /* reserved                     */
        unsigned char    format;          /* format                       */
        unsigned short   primary_rc;      /* primary return code          */
        unsigned long    secondary_rc;    /* secondary return code        */
        unsigned short   active;          /* status of the Attach Manager */
} QUERY_AM;
```

提供N数

opcode

AP_QUERY_ATTACH_MGR

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

返回N数

如果动词I功执行，则返回下P N }:

primary_rc

AP_OK

active C字段(f ，S管理器组件D状，:

AP_YES

，S管理器G活动D。

AP_NO

，S管理器G ；活动D。

如果因N }错误而9动词；能执行，则返回下P N }:

primary_rc

AP_PARAMETER_CHECK

如果因未启动Z c而9动词；能执行，则，S管理器返回下P N }:

primary_rc

AP_NODE_NOT STARTED

如果因系统错误而9动词；能执行，则，S管理器返回下P N }:

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

第2？分 个人通E和通E服务器管m服务 API

© Copyright IBM Corp. 1989, 1997, 1998

第13章 管m服务 API 的i \

> ? 分 h v K v 人通信或通信服务器 a 供D管理服务 API。

管m服务动词

v 人通信或通信服务器支V下P管理服务(MS)动词，为&C L 序 a 供K (f 1 在问b x SNA 网g中I C管理服务 9 c D方法。

- ALERT_INDICATION
- FP_INDICATION
- MDS_MU_RECEIVED
- NMVT_RECEIVED
- SEND_MDS_MU
- TRANSFER_MS_DATA

入口点

v 人通信或通信服务器 a 供K处理管理服务动词D b 文件。

管理服务动词P直S D o 言g f 。您D L 序<] F 为*动词X制 i* D内存 i 中D字段。然后L序w C入Z c " 传] 指k至动词X制 i 。C Y作完I后，管理服务(MS)API 返回以 9 C和修D动词X制 i 中D字段。L序M I 以读取从动词X制 i 返回D N } 。下P y > G管理服务动词D入Z c P m:

- ACSSVC()
- WinMS()
- WinAsyncMS()
- WinAsyncMSEx()
- WinCSV()
- WinMSCancelAsyncRequest()
- WinMSCleanup()
- WinMSStartup()

k N阅Z591页D『Z14B 管理服务入Z c 』以获取入Z c D详细 5 w。

动词控制块(VCB)

程 r h F " b: 基 > Y作系统通过在w C & C L 序D X址U间中执行一些子系统来E化性能。b意味着未完全或 } 确w T D & C L 序对 > X h v 符m(LDT)选择器D ; } 确 9 C + < 致 ; j 准D Y作，也I能引起系统 ' \ 。 相&D，&C L 序 ; &C执行N k | D指k D LDT 选择器字段D指k c u 运c 。

© Copyright IBM Corp. 1989, 1997, 1998

动词X制 i (VCB) y 9 C D段X须G I 读 I 写D }] 段。您D L 序 I 能Q VCB 5 w为L 序中D 一 v d ？来分配 |，也 I 能从一 v O大D段再次分配 |。 | X须大 = 足以 | 含L 序 } 在发 v D动词D y P字段。

& C L 序；& C 在发 v 动词X 制 i 后， | D动词X 制 i D任何？分，直至动词已完 I。管理服务 a x 执行动词后， 4 制一 v 完 { D，修D过D VCB = 原来D i O。 y 以，如果您D L 序Q 动词X 制 i 5 w为一 v d ？， k 在 2 ，存储器中 5 w，而；要在内？过L D堆；中 5 w。

C 0(X'00') n d ？ v VCB 中D y P # t 和未 9 C D字段。B 5 O，在L 序 3 值 x N } 之 0 h 置 { v 动词X 制 i 为 c 也许在 1 间O | 为P 效。 h 置 # t D字段为 c X p 重要。

"： 如果 VCB ；G I 读 I 写D，或 _ ；G至Y P 10 v 字Z (也MG，大 = 足够 # V管理服务主要和次级D返回 k)，管理服务M；能访问 |，而R 基 > Y 作系统 + 异 # 终止 x L。 b 种终止 j 6 为一 *c # 护故O*，处理器异 # i v 陷 e D。

管理服务在 VCB 过短或 9 C K ； } 确D段类型 1，返回 INVALID_VERB_SEGMENT 主要返回 k。

编 4 管m服务(MS)程 r

v 人通信或通信服务器 a 供动，4 S b (DLL)文件，以处理管理服务动词。

DLL G I 重入D；多 v & C L 序 x L 和线L I 以同 1 w C DLL。

管理服务动词P 直 S D o 言 g f 。您D L 序 <] F 为*动词X 制 i* (VCB) D内存 i 中D字段。然后，| w C管理服务 DLL " 传] 一 v 指 k = 动词X 制 i 。 C Y 作完 I 后，管理服务返回，以 9 C 和修D VCB 中D字段。 L 序M I 以读取从动词X 制 i 返回D N } 。

Z 588页D m 3 显 > y a 供头文件D源L 序模 i C法和需要来 ` 译、 4 S 管理服务L 序 D b 。某些头文件 I 能 | 括其 | y 需D头文件。

m 3. 头文件和管理服务 b

Y w **系统**	**头文 ~**	**库**	**DLL 名称**
WINNT & WIN95	WINMS.H	WINMS32.LIB	WINMS32.DLL
WIN3.1	WINCSV.H	WINCSV.LIB	WINCSV.DLL
OS/2	ACSSVCC.H	ACSSVC.LIB	ACSSVC.DLL

SNA API 客户机支持

通信服务器 | 括一系 P 基 Z Windows 95、Windows NT、Windows 3.1 和 OS/2 Y 作系统D M户机。 b 些M户机在 > i 中 F 为 SNA API M户机，而R v 支V全？管理服务动词D 一 v 子集。 _ e 如下:

- **WINMS** G v 在 Windows 95 和 NT M户机O 支V D API， k N 阅 Z 594页 D 『WinMS() 』以获取详细信息。

- **WINCSV** v在 Windows 3.1 M户机O支V， k N阅 Z593页D『WinCSV()』以获取详细信息。
- **ACSSVC** v在 SNA API OS/2 M户机O支V， k N阅 Z592页D『ACSSVC()』以获取详细信息。

下P y > G y支V D管理服务动词D P m:

- TRANSFER_MS_DATA
- SEND_MDS_MU

第14章 管m服务入口点

> B h v管理服务动词D人Z c。

© Copyright IBM Corp. 1989, 1997, 1998

ACSSVC()

b G支V SNA API OS/2 M户机D唯一入Z c。

b 为在 OS/2 SNA API M户机O发v D下P管理服务 API 动词a 供K同=入Z c。

o 法

```
void ACSSVC (long);
```

d 入D G动词X制 i 指k。

返回

为返回值检 i 主要和次级返回 k。

WinCSV()

b G支V Windows 3.1 M户机D唯一入Z c。

C函}为 CSV API a供同=入Z c。

o法

```
void WINAPI WinCSV(long vcb)
```

N数 5明

vcb 指向动词X制i D指k。

返回

无返回值。动词X制i中D **primary_rc** 和 **secondary_rc** 字段指v任何错误。

WinMS()

WinMSCleanup()

C函}从管理服务 API 终止"注销&CL序。

o法

```
BOOL WINAPI WinMSCleanup(void);
```

返回

返回值指定注销G否I功。如果值；为c， 则&CL序I功注销。如果返回值为c，则&CL序未；注销。

备"

9C **WinMSCleanup()** 来指v管理服务&CL序从管理服务 API D注销。

WinMSCleanup bi在 **WinMSGetIndication** 中D任何线LH待。带P WMSNOTREG返回(&CL序；注a来SU指>)。**WinMSCleanup** 为y P指>注销&CL序。**WinMSCleanup** 返回某些带P错误 AP_CANCELLED D未完I动词(同=或异=)。然而，动词G在Z c内完ID。

；需要9C **WinMSStartup** 和 **WinMSCleanup**。然而，&CL序X须在9Cb些wC方f #V一致。您&C=v都9C或=v都；9C。

"：m见 **WinMSStartup()**。

WinMSStartup()

C函 } 允许& C L 序指定 y 需D管理服务 API D f > " 检w z 品支V D API D f >。
C函 } I 以在发 v 任何 x 一 = D管理服务 API w C至寄存器之 0 I & C L 序来w C。

o 法

```
int WINAPI WinMSStartup(WORD wVersionRequired,
                        LPWMSDATA msdata);
```

N数 5 明

wVersionRequired

指定 y 需D管理服务 API 支V D f >。 _ 位字 Z 指定次 f >(revision)号; M位字 Z 指定主 f > 号。

msdata

返回管理服务 API D f > 和管理服务 5 现D 5 w。

返回

返回值指定& C L 序G否已 I 功注 a 和管理服务 API 5 现G否 I 以支V指定D f > 号。如果值为 c , 则为注 a I 功R I 以支V指定D f >。否则, 返回值G下 P 值之一:

WMSSYSERROR

基 > 网 g 子系统未准 8 好 x 行网 g 通信。

WMSVERNOTSUPPORTED

C X定管理服务 API 5 现未 a 供 k s D管理服务 API 支V D f >。

WMSBADPOINTER

; } 确D msdata N } 。

备 "

WinMSStartup 旨在 P 助 Z API + 来 f > D兼容性。支V D 1 0 f > G 1.0。

;需要 9 C **WinMSStartup** 和 **WinMSCleanup**。然而, & C L 序X须在 9 C b 些w C 方 f # V 一致。您& C = v 都 9 C或 = v 都 ; 9 C。

" : m见 **WinMSCleanup()**。

WinMSRegisterApplication()

b v函 } Q & C L序注 a 为 NMVT 级 & C L 序，MDS 级 & C L 序或 (/ 处理器。 b 样 D注 a v 定K & C L 序 S U哪 v 非 k s 指 > 。

- NMVT 级 & C L 序 S U NMVT_RECEIVED 指 > 。
- MDS 级 & C L 序 S U MDS_MU_RECEIVED 指 >， 1 9 c 状，D d 1，也 S U FP_NOTIFICATION 指 > 。
- (/ 处理器 S U ALERT_INDICATION 指 > 。

": 通过转换为 MDS MU 也 I 以注 a I S U NMVT。

; 处理 b 些指 > D & C L 序；& C w C **WinMSRegisterApplication**。

o 法

```
BOOL WINAPI WinMSRegisterApplication(unsigned short reg_type,
                                     unsigned char  *ms_appl_name,
                                     unsigned short vector_key,
                                     unsigned char  mds_conv_reqd,
                                     unsigned char  *ms_category,
                                     unsigned short max_rcv_size,
                                     unsigned char  alert_dest,
                                       unsigned short *primary_rc,
                                       unsigned long *secondary_rc);
```

N数 5明

reg_type

Registration type

WMSNMVTAPP	NMVT-level application (or MDS-level application registering to receive NMVTs)
WMSMDSAPP	MDS-level application
WMSALERTHANDLER	Alert handler

ms_appl_name

管理服务 & C L 序 { F。 P 效 { F I 以G 8 v 字 Z D字母 } 字D类型 1134 EBCDIC 字符串， X要D话 n O尾 f D U q (X'40')字符，或 *SNA 管理服务N* < 大全 = < D 中指定D管理服务规则指定 & C L 序之一，再 n O尾 f U q (X'40') 字符。

b v { F在 **reg_type** G WMSNMVTAPP 或 WMSMDSAPP 1 9 C。C { F ; J C Z **reg_type** G WMSALERTHANDLER 1 。

vector_key

& C L 序允许D值 S \ D管理服务主向 ? 关键字G:

X'YYYY'	specific major vector key
AP_SPCF_KEYS	major vector keys X'8061' through X'8064'
AP_ALL_KEYS	all major vector keys

b v关键字在 **reg_type** G WMSNMVTAPP 1 9 C。C关键字；J C Z **reg_type** G WMSMDSAPP 或 WMSALERTHANDLER 1 。

WinMSRegisterApplication()

mds_conv_reqd

指定 } 注 a D & C L 序 G 否为 MDS 级 " 要 s 发 M 至 | D NMVT 转换为 MDS MU。

(AP_YES 或 AP_NO)

b v N } 在 **reg_type** G WMSNMVTAPP 1 9 C。C N } ; J C Z **reg_type** G WMSMDSAPP 或 WMSALERTHANDLER 1 。

ms_category

1 & C L 序要获取 t Z 某类 p D 9 c D 信息 1，指定管理服务 D 类 p。管理服务类 p I 以 G *SNA 管理服务 N* < 大全 = < D D 管理服务规则指定 & C L 序 m 中指定 D 类 p 代 k 之一，再 n O 尾 f U q (X'40') 字符或 G C 户定义 D 类 p。C 户定义 D 类 p { F & C G 8 v 字 Z D 字母 } 字 D 类型 1134 EBCDIC 字符串，X 要 D 话 n O 尾 f D U q (X'40') 字符。

b v N } 在 **reg_type** G WMSMDSAPP 1 9 C。C N } ; J C Z **reg_type** G WMSNMVTAPP 或 WMSALERTHANDLER 1 。

max_rcv_size

& C L 序 I 以在一段中 S U D 最大字 Z } 。H C 值大 D MDS MU + ; 分段，R ? v 段都 ; 分发 = % 独 D MDS_MU_RECEIVED 指 > 中去。

b v N } 在 **reg_type** G WMSMDSAPP 1 9 C。C N } ; J C Z **reg_type** G WMSNMVTAPP 或 WMSALERTHANDLER 1 。

alert_dest

指定 & C L 序 G 否希望 I 为 y P (/ D 唯一目 D X。如果 | h 置为 AP_YES，则 y P (/ + 7 I 至 & C L 序，而 ; 7 I = 其 { 任何 X 方。如果 h 置为 AP_NO，(/ + 7 I 至 & C L 序 " g 越通 # 方法中 D SNA 网 g。

b v N } 在 **reg_type** G WMSALERTHANDLER 1 9 C。C N } ; J C Z **reg_type** G WMSNMVTAPP 或 WMSMDSAPP 1 。

primary_rc

返回：主要返回 k

secondary_rc

返回：次级返回 k

返回

函 } 返回一 v 指 > 注 a G 否 I 功 D 值。如果值 ; 为 c，则注 a I 功。如果值为 c，则注 a ; I 功。

备 "

& C L 序 I x 行多 v w C 来注 a 多 v 级 p D 指 > 。

w C **WinMSRegisterApplication** D & C L 序 X 须 w C **WinMSGetIndication** 来 S U 为其排队 D 指 > 。

" : m 见 **WinMSUnregisterApplication** 和 **WinMSGetIndication**。

WinMSUnregisterApplication()

b v 函 } 注销 & C L 序， k 先 0 D **WinMSRegisterApplication** w C 效果相反， " 停止为 & C L 序排队 D x 一 = 指 > 。

o 法

```
BOOL WINAPI WinMSUnregisterApplication(unsigned short reg_type,
                                       unsigned char  *ms_appl_name,
                                         unsigned short *primary_rc,
                                         unsigned long *secondary_rc);
```

N数　5 明

reg_type

注 a 类型。 | I 以 G 下 P 值之一:

WMSNMVTAPP
NMVT 级 & C L 序

WMSMDSAPP
MDS 级 & C L 序

WMSALERTHANDLER
(/ 处理器

ms_appl_name

MS & C L 序 { F。 P 效 { F I 以 G 8 v 字 Z D 字母 } 字 D 类型 1134 EBCDIC 字符串， X 要 D 话 n O 尾 f D U q (X'40') 字符，或 *SNA 管理服务 N* < 大全 = < D 中指定 D 管理服务规则指定 & C L 序之一，再 n O 尾 f U q (X'40') 字符。

b v N } 在 **reg_type** G WMSNMVTAPP 或 WMSMDSAPP 1 9 C。 C N } ; J C Z **reg_type** G WMSALERTHANDLER 1 。

primary_rc

返回: 主要返回 k

secondary_rc

返回: 次级返回 k

返回

函 } 返回一 v 指 > 注销 G 否 I 功 D 值。如果值 ; 为 c， 则注销 I 功。如果值为 c， 则注销 ; I 功。

备 "

? v 对 **WinMSUnregisterApplication** D w C 终止 O 早 0 对 **WinMSRegisterApplication** D w C 1 y x 行 D 注 a 。多次 w C **WinMSRegisterApplication** D & C L 序需要多次 w C **WinMSUnregisterApplication** 来终止 y P 注 a 。

WinMSUnregisterApplication 和 **WinMSCleanup** D ; 同之处在 Z:

- **WinMSUnregisterApplication** 终止先 0 注 a 来 S U 指 >， + ; 能阻止 & C L 序 x 行其 | 管理服务 API w C (例如， WinMS)。

WinMSUnregisterApplication()

- **WinMSCleanup** 终止K管理服务 API D 9 C。

1 & C L 序w C **WinMSUnregisterApplication** 1 I 能已 - P 指 > 为& C L 序排队K。任何b 样D指 > 都 # V 排队， " R & C L 序& C w C **WinMSGetIndication** 来S U 和处理 | G。一) | G ；注销， M ； P 新D指 > 为& C L 序排队K。

"： m见 **WinMSRegisterApplication** 和 **WinMSGetIndication**。

WinMSGetIndication()

b 允许& C L 序 S U非 k s D指 > 。

o 法

```
int WINAPI WinMSGetIndication(long buffer,
                              unsigned short *buffer_size,
                              unsigned long timeout);
```

N数　5 明

buffer 指向C Z S U指 > D缓 e x D指 k 。

buffer_size
缓 e x 大小。返回: 指 > D大小。

timeout
以毫 k 为%位D H待指 > 1 间。

返回

函 } 返回一 v 值, 5 w G否 S U = 指 > 。

0 返回D指 > 。

WMSTIMEOUT
H待指 > D , 1 。

WMSSYSNOTREADY
基 > 网 g 子系统未准 8 好 x 行网 g 通信。

WMSNOTREG
& C L 序未注 a S U指 > 。

WMSBADSIZE
缓 e x + 小, ; 能 S U指 > 。带 P 足够大D缓 e x 再次发 v **WinMSGetIndication** w C。指 > D大小返回 = **buffer_size** N } 中。

WMSBADPOINTER
缓 e x 或 **buffer_size** N } 无效。

WMSSYSERROR
发 z 未预O = D系统错误。

备 "

b G一 v 阻塞w C, | 在 v 现下 P i v 之一 1 返回:

- 返回一 v 指 >
- , 1 = 期
- & C L 序发 v **WinMSCleanup** w C
- z 品停止K
- 发 z 系统错误

WinMSGetIndication()

"： m见 **WinMSRegisterApplication** 和 **WinMSUnregisterApplication**。

第15章 管m服务动词

v 人通信或通信服务器 a 供D管理服务 API 动词启C一v & C L 序来发M / (和 MDS MU }] , " 1 Z c S U = MDS 或 NMVT }] 及发v / (1 S U指 > 。

© Copyright IBM Corp. 1989, 1997, 1998

TRANSFER_MS_DATA

NMVT 级&C L序9 C C动词来发M未 - k s D / (, "响&先0已U = D NMVT k s。

MDS 级&C L序也I以9 C TRANSFER_MS_DATA 来发M未 - k s D / (。9 C WinMS w C D&C L序也I以9 C C动词。

VCB a构

```
typedef struct ms_transfer_ms_data
{
   unsigned short   opcode;              /* Verb operation code       */
   unsigned char    data_type;           /* Data type supplied by app */
   unsigned char    format;              /* format                    */
   unsigned short   primary_rc;          /* Primary return code       */
   unsigned long    secondary_rc;        /* Secondary return code     */
   unsigned char    options;             /* Verb options              */
   unsigned char    reserv3;             /* reserved                  */
   unsigned char    originator_id[8];    /* Originator ID             */
   unsigned char    pu_name[8];          /* Physical unit name        */
   unsigned char    reserv4[4];          /* reserved                  */
   unsigned short   dlen;                /* Length of data            */
   unsigned char    *dptr;               /* Data                      */
} MS_TRANSFER_MS_DATA;
```

提供的N数

&C L序a供下P N }:

opcode

SV_TRANSFER_MS_DATA

data_type

指定y含 }] D类型。管理服务4如下 y v来处理 }]。允许值为:

SV_NMVT

C }] | 含一v完{ D NMVT k s %元。如果 }] | 含一v / (, R / (+要;发M至 MDS 级或(移级9 c, 则管理服务会+ }]转换I MDS_MU 或 CP_MSU q =。b G 1 &C L序响&一v NMVT_RECEIVED 信号1需要D类型。

SV_ALERT_SUBVECTORS

C }] | 含 / (主向?D SNA 定义q = D管理服务子向?。管理服务m加一v NMVT W?和一v / (主向?W?。S着, 如果 / (+要;发M至 MDS 级或(移级9 c, 管理服务会+ }]转换I MDS_MU 或 CP_MSU q =。

SV_USER_DEFINED

C }] | 含一v完{ D NMVT k s %元。管理服务总G记< }], +;发M |。

SV_PDSTATS_SUBVECTORS

C }] | 含问b确定统计。管理服务总G记< }], R如果已注a K / (处理器, 管理服务+传M x | ALERT_INDICATION 内D }].

format

j 6 VCB D q = 。 + C字段 h 置为 c，以指定O f P v D VCB D f > 。

options

指定C动词 a 供D }] O D I 选处理。注意，管理服务主要 y] **data_type** 和指定选项之间发 z e 突 1 指定D **type** 来处理 }] 。 C N } G 一 v %字 Z 值，其 v p 位 h 置 m > 选定D选项。如果指定K y P D选项，则 + C字 Z h 置为 c 。

Z 0 位G最重要D，而Z 7 位G最；重要D位。

如果 **data_type** h 置为 SV_USER_DEFINED，则忽T Z 1-3 位。)

Z 0 位: 如果 h 置为 c，则 + 日期/ 1 间(X'01')子向？m加至 }] 。

Z 1 位: 如果 h 置为 c，则 + z 品 h 置 ID (X'10')子向？m加至 }] 。如果 & C L 序 a 供K已 | 含 z 品 h 置 ID 子向？D }]，则管理服务会 " 即在已存在D子向？之 0 m加 v 人通信或通信服务器D z 品 h 置 ID 子向？。

Z 2 位: 如果 h 置为 c，则 + }] 发M至 SNA 会话。如果 }] ；| 含 / (，管理服务会 + }] 发M至缺！ SSCP-PU 会话。如果 }] | 含 / (， y] v 人通信或通信服务器向 / (9 c 发M / (1 9 C D会话类型，管理服务 + }] 发M x SSCP-PU 会话、CP-CP 会话或 LU-LU 会话。

Z 3 位: 如果 h 置为 c，则 - I v 人通信或通信服务器问 b 确定 h) 来记 < }] 。

" **:** 在管理服务头文件中 a 供下 P # } ， R | G N < O f y 指定D v p 位。

- SV_TIME_STAMP_SUBVECTOR
- SV_PRODUCT_SET_ID_SUBVECTOR
- SV_SEND_ON_SESSION
- SV_LOCAL_LOGGING

Z 4-7 位: # t 。

originator_id

发 v 动词D组件 { 。 | G > X I 显 > 字符集中D一 v 8 字 Z 字符串。 1 记 < TRANSFER_MS_DATA 1， C字段 v I 管理服务 9 C。

pu_name

要向之发M }] D物理%元 { 。 & C + 其 h 置为一 v 8 字 Z 字母 } 字D A 型 EBCDIC 字符串， " 以 EBCDIC U q R n d，或如果；P 指定 **pu_name**，则 h 置为全二 x 制 c 。 9 C TRANSFER_MS_DATA 响& NMVT_RECEIVED 信号D & C L 序& C 指定 NMVT_RECEIVED 信号中 S U = D **pu_name**。；P 指定 **pu_name** D类型 SV_NMVT D TRANSFER_MS_DATA 信号中 | 含D }] 如果 I C D 话 + 在缺！ PU 会话 O ；发M。 | 含 / (D TRANSFER_MS_DATA 信号；& C 指定 **pu_name** ， } 非& C L 序 X p 希望 + / (}] 发M = X 定D PU。 b + \ * } # D管理服务 / (7 I 选择 c 法。

dlen }] D $ 度。

dptr 指向 }] D指 k 。如果 h 置为 NULL，那 4 管理服务会假 h }] k VCB 相，(" " 即 * < z 踪)。

TRANSFER_MS_DATA

返回的N数

如果动词 I 功执行，管理服务 + 返回下 P N }:

primary_rc
AP_OK

如果动词因 N } 错误而执行 ' \ ，管理服务 + 返回下 P N }:

primary_rc
AP_PARAMETER_CHECK

secondary_rc
SV_INVALID_DATA_TYPE

SV_DATA_EXCEEDS_RU_SIZE
AP_INVALID_PU_NAME

如果动词因状，错误而执行 ' \ ，管理服务 + 返回下 P N }:

primary_rc
AP_STATE_CHECK

secondary_rc
SV_SSCP_PU_SESSION_NOT_ACTIVE

如果动词因系统错误而；能执行， L 序管理服务 + 返回下 P N }:

primary_rc
AP_UNEXPECTED_SYSTEM_ERROR

MDS_MU_RECEIVED

1 v现下P i v 1，管理服务+C动词指>发M至已注a D MDS 级&C L序:

- 已从同级 MDS 级&C L序S U= MDS_MU
- 已S U= NMVT， " R
  - ；P注a J 1 D NMVT 级&C L序
  - MDS 级&C L序注a D{ F k x入D NMVT(管理服务执行从 NMVT = MDS_MU D转换)中管理服务主向？\钥内携带D{ F相对&。

VCB a构

```
typedef struct ms_mds_mu_received
{
   unsigned short   opcode;          /* Verb operation code          */
   unsigned char    reserv2;         /* reserved                     */
   unsigned char    format;          /* format                       */
   unsigned short   primary_rc;      /* Primary return code          */
   unsigned long    secondary_rc;    /* Secondary return code        */
   unsigned char    first_message;   /* First message for curr MDS_MU */
   unsigned char    last_message;    /* Last message for curr MDS_MU  */
   unsigned char    pu_name[8];      /* Physical unit name           */
   unsigned char    reserv3[8];      /* reserved                     */
   unsigned short   mds_mu_length;   /* Length of incoming MDS_MU    */
   unsigned char    *mds_mu;         /* MDS_MU data                  */
} MS_MDS_MU_RECEIVED;
```

提供的N数

opcode

AP_MDS_MU_RECEIVED

format

j 6 VCB D q=。C字段h置为c以5 w以O P v D VCB f >。

first_message

j志指v b G；G MDS_MU (AP_YES 或 AP_NO) D Z一u信息。如果 **WinMSRegisterApplication** w C中指定D **max_rcv_size** H传]中 MDS_MU D $度小，则以组i形=+ MDS_MU 发M至&C L序。

last_message

j志指v b G；G MDS_MU (AP_YES 或 AP_NO) D最后一u信息。

pu_name

<v NMVT (已转换为 MDS_MU) D物理%元{。I&C L序：责响&x入D NMVT。C&C L序9C SEND_MDS_MU 发M响&。1发M响&1， &C L序X须+ SEND_MDS_MU D **pu_name** 字段h置为 MDS_MU_RECEIVED 信号中a供D **pu_name**。如果 MDS_MU G从 MDS 级传M机制S U=D，则 **pu_name** +；h置为全二x制c。

mds_mu_length

k信号|括在一起D MDS_MU ？分D $度。

mds_mu

MDS_MU }]。}]指k h置为 NULL， R}] k VCB 相，(" "即*<z踪)。

SEND_MDS_MU

MDS 级&C L序9 C C动词来发M网g管理 }]，而；发M 9 C **WinMS** 入Z c D /（。如果在发M MDS_MU 至目D&C L序期间发z K错误，则错误+以一种或=种方=；（f回起<&C L序。如果在>X Z c处检b=错误，则+ - I SEND_MDS_MU响&D返回k来通知&C L序。如果在远L Z c处检b=错误，则+通过MDS_MU_RECEIVED VCB 中传M D错误 MDS_MU 来（f错误。如果要- I SSCP-PU 会话=达目D Z c，则管理服务能+v网 MDS_MU 转换为 NMVT。&C L序；X知@|>X Z c D m份。如果&C L序在 MDS 7 I选择信息 GDS d ? D起<位置{子向？D **netid** 或 **nau** 或同1在=v子字段中a供K 8 v EBCDIC U q，则v人通信或通信服务器+a供J 1 D值。如果&C L序；P在 **netid** 或 **nau** 中n d，却a供K小Z 8 v D U q，则v人通信或通信服务器+返回AP_INVALID_MDS_MU_FORMAT D次级返回k。

VCB a构

```
typedef struct ms_send_mds_mu
{
   unsigned short   opcode;             /* Verb operation code         */
   unsigned char    reserv2;            /* reserved                    */
   unsigned char    format;             /* format                      */
   unsigned short   primary_rc;         /* Primary return code         */
   unsigned long    secondary_rc;       /* Secondary return code       */
   unsigned char    options;            /* Verb options                */
   unsigned char    reserv3;            /* reserved                    */
   unsigned char    originator_id[8];   /* Originator ID               */
   unsigned char    pu_name[8];         /* Physical unit name          */
   unsigned char    reserv4[4];         /* reserved                    */
   unsigned short   dlen;               /* Length of data              */
   unsigned char    *dptr;              /* Data                        */
} MS_SEND_MDS_MU;
```

提供的N数

opcode

AP_SEND_MDS_MU

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

options

指定C动词a供D }] O D任选处理。C N } G一v %字Z值，C v p位h置指定选定D选项。如果指定K y P D选项，则+C字Z h置为c。

Z 0 位G最重要D，而Z 7 位G最；重要D位。

Z 0 位：如果h置为c，则+日期/1间(X'01')子向？m加至 }]。

Z 1 位：如果h置为c，则+z品h置 ID (X'10')子向？m加至 }]。如果&C L序a供K已|含一v z品h置 ID 子向？D }]，那4管理服务会"即在现存子向？0 m加v人通信或通信服务器D z品h置 ID 子向？。

Z 2 位：# t。

Z 3 位：如果h置为c，则- I v人通信或通信服务器问b确定h)来记<}]。

": 在管理服务头文件中 a 供下 P # }, R | G引|C O f y 指定D Z 0、1 和 3 位。

- SV_TIME_STAMP_SUBVECTOR
- SV_PRODUCT_SET_ID_SUBVECTOR
- SV_LOCAL_LOGGING

Z 4 位: 指定管理服务G要 9 C缺！还G直S 7 I + 管理服务 }] 发M至目 D & C L 序(AP_DEFAULT 或 AP_DIRECT)。

": 要 h 置Z 4 位, 9 C J 1 移位D AP_DEFAULT 或 AP_DIRECT (例如, AP_DIRECT<<3)。

Z 5-7 位: # t 。

originator_id
发 v 动词D组件 { 。 1 记 < SEND_MDS_MU 1, C字段 v I 管理服务 9 C。

pu_name
要向之发M }] D物理%元 { 。 & C + 其 h 置为一 v 8 字 Z 字母 } 字D A 型 EBCDIC 字符串, " 以 EBCDIC U q R n d, 或如果 ; P 指定 **pu_name**, 则 h 置为全二 x 制 c 。 9 C SEND_MDS_MU 响&从 x 入D NMVT 转换来D MDS_MU_RECEIVED 指 > D & C L 序& C 指定 MDS_MU_RECEIVED 中 S U = D **pu_name**。要 9 C MDS 传M h) 来传M D MDS_MU & C h 置 **pu_name** 为全二 x 制 c 。

dlen }] D $ 度。

dptr 指向 }] D 指 k 。如果 h 置为 NULL, 管理服务会假 h }] k VCB 相 , (" " 即 * < z 踪)。

返回的N数

如果动词 I 功执行, L 序管理服务 + 返回下 P N } :

primary_rc
AP_OK

如果动词因 N } 错误而执行 ' \ , L 序管理服务 + 返回下 P N } :

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_PU_NAME

AP_INVALID_MDS_MU_FORMAT
SV_INVALID_DATA_SIZE

如果动词因状，错误而执行 ' \ , L 序管理服务 + 返回下 P N } :

primary_rc
AP_STATE_CHECK

secondary_rc
AP_SSCP_PU_SESSION_NOT_ACTIVE

如果动词因系统错误而；能执行， L 序管理服务 + 返回下 P N }：

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

ALERT_INDICATION

管理服务 9 C C 动词指 > + / (主向 ? 发 M 至 + 要处理 | G D 已注 a / (处理器或已注 a 挂起 / (处理器。

VCB a 构

```
typedef struct ms_alert_indication
{
   unsigned short   opcode;             /* AP_AlERT_INDICATION         */
   unsigned char    reserv2;            /* reserved                    */
   unsigned char    format;             /* format                      */
   unsigned short   primary_rc;         /* Primary return code         */
   unsigned long    secondary_rc;       /* Secondary return code       */
   unsigned short   alert_length;       /* Length of alert             */
   unsigned char    reserv3[6];         /* reserved                    */
   unsigned char    *alert;             /* Alert data                  */
} MS_ALERT_INDICATION;
```

提供的 N 数

opcode

AP_ALERT_INDICATION

format

j 6 VCB D q = 。 + C 字段 h 置为 c , 以指定 O f P v D VCB D f > 。

alert_length

/ (}] D $ 度。

alert 指向 / (}] D 指 k 。 }] 指 k h 置为 NULL, R }] k VCB 相 , (" " 即 * < z 踪)。

FP_NOTIFICATION

如果 MDS 级&C L序已为X定管理服务类p注 a R C类p D 9 c状，| D K，则管理服务 + C动词信号发M至&C L序。

VCB a构

```
typedef struct ms_fp_notification
{
   unsigned short   opcode;             /* Verb operation code         */
   unsigned char    reserv2;            /* reserved                    */
   unsigned char    format;             /* format                      */
   unsigned short   primary_rc;         /* Primary return code         */
   unsigned long    secondary_rc;       /* Secondary return code       */
   unsigned char    fp_routing;         /* Type of routing to focal pt */
   unsigned char    reserv1;            /* reserved                    */
   unsigned short   fp_data_length;     /* Length of incoming focal    */
                                        /* point data                  */
   unsigned char    *fp_data;           /* focal point data            */
} MS_FP_NOTIFICATION;
```

提供的N数

opcode

AP_FP_NOTIFICATION

format

j 6 VCB D q =。+ C字段 h 置为 c，以指定O f P v D VCB D f >。

fp_routing

1 发M信息至 9 c (AP_DEFAULT 或 AP_DIRECT) 1， &C在 SEND_MDS_MU O指定D 7 I 选择D类型。

fp_data_length

9 c }] D $ 度。

fp_data

| 含 9 c 通知(X'E1')子向？和 9 c j 6 (X'21')子向？D 9 c }]。}] 指 k h 置为 NULL， R }] k VCB 相，(" " 即 * < z 踪)。

NMVT_RECEIVED

1 从远L Z c S U = NMVT 1，管理服务会+C动词发M至已注a D NMVT 级&C L序。

在7 I x入D NMVT 1，管理服务J C下P规则:

1. " T 7 I至9 C x入D NMVT O携带D主向？\ 钥注a D NMVT 级&C L序，或 _ ...
2. 如果主向？\ 钥G X'8061' = X'8064' 中D一v，则" T 7 I至已注a D NMVT 级 AP_SPCF_KEYS &C L序，或 _ ...
3. " T 7 I至 NMVT 级已注a D AP_ALL_KEYS &C L序，或 _ ...
4. " T + NMVT(转换为 MDS_MU 后)7 I至9 C x入D NMVT O携带D主向？\ 钥注a D NMVT 级&C L序，或 _ ...
5. 如果主向？\ 钥G X'8061' = X'8064' 中D一v，则" T + NMVT(转换至 MDS_MU 后)7 I至已注a D MDS 级&C L序，或 _ ...
6. " T 7 I(转换至 MDS_MU 后)至已注a D AP_ALL_KEYS MDS 级&C L序，或 _ ...
7. 否定响& NMVT。

VCB a构

```
typedef struct ms_nmvt_received
{
   unsigned short   opcode;             /* Verb operation code         */
   unsigned char    reserv2;            /* reserved                    */
   unsigned char    format;             /* format                      */
   unsigned short   primary_rc;         /* Primary return code         */
   unsigned long    secondary_rc;       /* Secondary return code       */
   unsigned char    pu_name[8];         /* Physical unit name          */
   unsigned char    reserv3[6];         /* reserved                    */
   unsigned short   nmvt_length;        /* Length of incoming NMVT     */
   unsigned char    *nmvt;              /* NMVT data                   */
} MS_NMVT_RECEIVED;
```

提供的N数

opcode

AP_NMVT_RECEIVED

format

j 6 VCB D q =。+C字段h置为c，以指定O f P v D VCB D f >。

pu_name

< v NMVT D物理%元{。I&C L序：责响& x入D NMVT 。C&C L序 9 C TRANSFER_MS_DATA 来发M响&。1发M响& 1，&C L序X须+ TRANSFER_MS_DATA 中D **pu_name** 字段h置为 NMVT_RECEIVED 信号 衷猪吱诛逐 h 蛛皱蜘衷株织制止值 直 衷宙殖之侄职之职直殖 衷帚吱吱枝蜘衷株殖制

附 < A. IBM APPN MIB 表

下 m x v K 如 RFC1593 y 定义 D 从 IBM APPN 管理信息 i (MIB) 5 现 m D 细 Z。 C m 定义:

- Z c Y 作员 h) QUERY 动词 C Z 5 现 ? v MIB m
- d 人 N } h 置
- 任何 X 需 D 过 K Y 作

9(返回 D N } 和 MIB m d ? 间 D 3 象 I 以从 Z c Y 作员 h) QUERY 动词取 C)。 v 人通信或通信服务器 1 0 ; 支 V ibmappnNodePortDlcTraceTable 和 ibmappnLsStatusTable MIB m。

IBM MIB 表	Z 点 Y w 员 h **施动词和 MIB 表变** ?	**输入 N 数 h 置**
ibmappnNodePortTable	QUERY_PORT	**port_name** ibmappnNodePortName
ibmappnNodePortIpTable	(注 1)	
ibmappnNodePortDlsTable	QUERY_PORT (9 C AP_SDLC D **dlc_type** 来选择项)	**port_name**ibmappnNodePortDlsName
ibmappnNodePortTrTable	QUERY_PORT	**port_name** ibmappnNodePortTrName
ibmappnNodeLsTable	QUERY_LS	**ls_name** ibmappnNodeLsName
ibmappnNodeLsIpTable	(注 1)	
ibmappnNodeLsDlsTable	QUERY_LS (9 C AP_SDLC D **dlc_type** 来选择项)	**ls_name** ibmappnNodeLsDlsName
ibmappnNodeLsTrTable	QUERY_LS	**ls_name** ibmappnNodeLsTrName
ibmappnNnTopoRouteTable	QUERY_COS	**cos_name** ibmappnNnTopoRouteCos
ibmappnNnAdjNodeTable	QUERY_ADJACENT_NN	**a d j _ n n c p _ n a m e** ibmappnNnAdjNodeAdjName
ibmappnNnTopologyTable	QUERY_NN_TOPOLOGY_NODE	**node_name** ibmappnNnNodeName **node_type** AP_LEARN_NODE **frsn** 0

© Copyright IBM Corp. 1989, 1997, 1998

IBM MIB 表	Z 点 Y w 员 h 施动词和 MIB 表变？	输入 N 数 h 置
ibmappnNnTgTopologyTable	QUERY_NN_TOPOLOGY_TG	**owner** ibmappnNnTgOwner **owner_type** AP_LEARN_NODE **dest** ibmappnNnTgDest **dest_type** AP_LEARN_NODE **tg_num** ibmappnNnTgNum **frsn** 0
ibmappnNnTopologyFRTable	QUERY_NN_TOPOLOGY_NODE	**node_name** ibmappnNnFRNode **node_type** AP_LEARN_NODE **frsn** ibmappnNnFRFrsn
ibmappnNnTgTopologyFRTable	QUERY_NN_TOPOLOGY_TG	**owner** ibmappnNnTgFROwner **owner_type** AP_LEARN_NODE **dest** ibmappnNnTgFRDest **dest_type** AP_LEARN_NODE **tg_num** ibmappnNnTgFRNum **frsn** ibmappnNnTgFRFrsn
ibmappnLocalTgTable	QUERY_LOCAL_TOPOLOGY	**dest** ibmappnLocalTGDest **dest_type** AP_LEARN_NODE **tg_num** ibmappnLocalTgNum
ibmappnLocalEnTable	QUERY_LOCAL_TOPOLOGY (9 C 唯一 D **dest** 来选择项) (注 2)	**dest** ibmappnLocalEnName **dest_type** AP_END_NODE **dest_type** AP_LEARN_NODE

IBM MIB 表	Z 点 Y w 员 h 施动词和 MIB 表变 ?	输入 N 数 h 置
ibmappnLocalEnTgTable	QUERY_LOCAL_TOPOLOGY (注 3)	**dest** ibmappnLocalEnTgOrigin **dest_type** AP_LEARN_NODE **tg_num** ibmappnLocalEnTgNum
ibmappnDirTable	QUERY_DIRECTORY_LU	**lu_name** ibmappnDirLuName
ibmappnCosModeTable	QUERY_MODE_TO_COS_MAPPING	**mode_name** ibmappnCosModeName
ibmappnCosNameTable	QUERY_COS	**cos_name** ibmappnCosName

"：

1. v 人通信或通信服务器；支 V IP 为 DLC 类型。
2. P 相同 **dest** D 项 I QUERY_LOCAL_TOPOLOGY 4 序，续排 P 。
3. ibmappnLocalEnTgTable 从，S D 末端 Z c D 透 S 来 i 4 TG(即作为来自末端 Z c D TG)。+ G，遵从 1 0 APPN e 系 a 构级 p D 网 g Z c v 为 TG > m 和 k | 直 S 相，D 末端 Z c 之间 D TG 存储末端 Z c TG 信息。因此 > m 中 D y P 项都 + ibmappnLocalEnTgDest h 置为 > X Z c { (ibmappnNodeCpName)。

附 < B. " b 事项

> 信息 J C Z 在@国 a 供 D z 品和服务。 IBM 在其 | 国家 I 能； P a 供 > 信息中 V [D z 品、服务或功能。 k i 询您 D > X IBM 代 m 处，获取您 D x r 内 1 0 I C D z 品和服务。任何对 IBM z 品、L 序或服务 D 引 C " ； 5 w 或 5 > 只能 9 C IBM z 品、L 序或服务。任何 _ P 同 H 功能 D z 品、L 序或服务，只要；V 犯 IBM D 知 6 z 权，都 I 以 C 来 f 代 IBM D z 品、L 序或服务。 + G，评估和验证非 IBM z 品、L 序或服务 I C 户自行：责。

对 Z > 文 5 中 f 及 D 内容 D O，IBM I 能 5 P 专利或未 v 专利 D j k。 a 供 > 文 5 " ； m > 允许您 9 C b 些专利。您 I 以以 i f 方 = + 许 I i 询发 M 至:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

P 关 + 字 Z (DBCS)信息 D 许 I 证 i 询， k k 您 D 国家 D IBM 知 6 z 权？E * 系或 + i 询发 M 至:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

下段；适 C Z " **国或任何当地法** I k **本条款；；致的国** R: 国际 L 业机器公 > 以『 AS IS』(即“原样”)方 = a 供 > v f 物， > v f 物； _ P 任何形 = D # #，无 [G w 确 D 还 G 隐 = D，其中 | 括 + ；V 限 Z 对 X 定目 D 之合法性、I 销 [性或 J 宜性 D 隐 = # #。一些国家；允许在 X 定 B 务中 \ x w 确 D 或隐 = D # #，因此 > y w I 能； J C Z 您。

> 信息 I 能会 | 含技 u O D； + 确性或！" 错误。此处 a = D 信息也会 f 1 | D; b 些 | D + ；合 " 至 v f 物 D 新 f > 中。IBM I 能 f 1 在 > v f 物 y h v D z 品和/或 L 序中作 D x 和/或 | D，而；作 y w。

为 K 以下目 D: （i）允许在独 " 创 (D L 序和其 | L 序（ | 括 > L 序）之间 x 行信息；换（ii）允许对已 - ；换 D 信息 x 行相互 9 C，而希望获取 > L 序 P 关信息 D 合法 C 户 k k 下 P X 址 * 系:

IBM Corporation
Department TL3B/062
P.O. Box 12195
Research Triangle Park, NC 27709-2195
U.S.A.

依 U J 1 D u n 和 u 件，其中 | 括在一些 i v 下需要 6 费，b 些信息或许 G I C D。

> 信息中 h v D X 许 L 序和 y P I C D X 许资 O G I IBM 4 U IBM M 户协议、国际 L 序 h 计许 I 证协议或任何 H 价 D 协议 a 供 D。

© Copyright IBM Corp. 1989, 1997, 1998

P 关非 IBM z 品 D 信息 G 从那些 z 品 D 供 & L 、 { G 发 < D y w 或其 | 公共 I C D 资源获 C D 。 IBM ; P b T 过那些 z 品, R ; 能确认性能、兼容性或任何关 * 非 IBM z 品 D 其 | y w D 准确度。 P 关非 IBM z 品 D 能 & 问 b & C 致函那些 z 品 D 供 & L 。

f 权许 I 证:

> 信息 | 含 C 源 o 言写 D 样 > & C L 序, b 些 & C L 序 5 w K w 种 Y 作平 (O D L 序 h 计技 u 。为 K * 发、 9 C 、 P ! * 销或分发遵从 Y 作平 ((样 > L 序在其 O i 写)D & C L 序 h 计 S Z D & C L 序, I 以以任何形 = ; C 6 费 x IBM 来 4 制、修 D 和分发 b 些样 > L 序。 b 些 5 例 " ; P 在 y P D u 件下 9 W b T 过。因此 IBM ; 能 # 证或 5 > b 些 L 序 D I ? 性、 I 服务性或功能。为 K * 发、 9 C 、 P ! * 销或分发遵从 IBM & C L 序 h 计 S Z D & C L 序, I 以以任何形 = ; C 6 费 x IBM 来 4 制、修 D 和分发 b 些样 > L 序。

? v 1 > 、 b 些样 > L 序 D 任何 ? 分或任何派 z D 工作, 都 X 须 | 括 4 如下 y > D 1 > 注意 B 项:

(c) (您公 > D { F) (年)。 b v 代 k D 一 ? 分来源 Z IBM Corp. 样 > L 序。(c) Copyright IBM Corp. d 人年 } 。 # t y P 权。

附 < C. L 标

下 P u o G@国和/或其 | 国家中 IBM 公 > D L j :

ACF/VTAM
Advanced Peer-to-Peer Networking
AFP
AIX
AnyNet
APPN
AS/400
AT
CICS
Common User Access
CUA
IBM
IMS
MVS
MVS/ESA
MVS/XA
NetView
Operating System/2
OS/2
OS/400
RACF
System/370
Virtual Machine/Enterprise Systems Architecture
VM/ESA
VTAM

其 | 公 > 、 z 品和服务 { (I 能 C + 星号(**)m >) I 能 G 其 | 公 > D L j 或服务 j 记。

C-bus G Corollary, Inc D L j 。

ActionMedia、LANDesk、MMX、Pentium 和 ProShare G@国和其 | 国家中 Intel 公 > D L j 或注 a L j 。

Java 和 HotJava G Sun 微系统公 > D L j 。

Microsoft、Windows、Windows NT 和 Windows 95 徽 j G Microsoft 公 > D注 a L j 。

UNIX G通过 X/Open P限公 > 专 E X许 D 在@国和其 | 国家注 a D L j 。

PC Direct G Ziff 通信公 > D L j , " I P 许 I 证 D IBM 公 > 9 C。

© Copyright IBM Corp. 1989, 1997, 1998

w }

> w引 4 汉 o 拼音，} 字，" 文字母和X b 字符 3 序排 P 。

[A]

[B]

[C]

[D]

© Copyright IBM Corp. 1989, 1997, 1998

[G]

[H]

[J]

[L]

[R]

[X]

[Y]

[Z]

A

C

D

E

F

H

I

L

M

N

P

Q

R

S

T

W

X

读者 b { 表

eNetwork 通 E 服务器 f 本 6.0 Windows NT f 和 eNetwork 个人通 E f 本 4.2 Windows 95 和 Windows** NT f**

系统管 m程 r h F

SC84-0693-00

姓 {

X 址

% 位及 ? E

g 话号 k

读者 b { 表
SC84-0693-00

k 沿此线
: 下或 [起

[起 " 封 Z　　请勿使 C 钉书机　　[起 " 封 Z

在此
y O
J 票

IBM Corporation
Information Development
Department CGMD / Bldg 500
P.O. Box 12195
Research Triangle Park, NC
27709-9990

[起 " 封 Z　　请勿使 C 钉书机　　[起 " 封 Z

SC84-0693-00

k 沿此线
: 下或 [起

Printed in China

SC84-0693-00