

DNR Directory Services

This chapter describes the C library directory services function `dirsrv()` and both C and assembler language definitions for DPL/dpl, the Directory Services Parameter List. It includes these sections:

- The `dirsrv()` Function

Describes the basic C library directory services function `dirsrv()`, which is used to request a service from the Domain Name Resolver (DNR).

- Directory Services Parameter List (DPL)

Includes the assembler language (DPL) and the C language (dpl) definitions for the Directory Services Parameter List (DPL).

The dirsrv() Function

The dirsrv() function is used to request a service from the Domain Name Resolver (DNR). The DNR subsystem maintains a data base of information locally and throughout the network. The dirsrv() function provides a general-purpose interface to this information.

Format Description

This is the format description for this function:

```
#include <ds.h>
int      dirsrv ( dplp )
struct dpl      *dplp;
```

Description

The dirsrv() function performs the request specified by the format of the DPL passed to it. In the processing of this request, the DNR subsystem is called to perform the actual request. Any errors detected by the DNR are returned to the caller as the return code (error) of this function. In addition to the general return code, the DPL return code field can be set to a value to more clearly distinguish the reason for the error. A return code of 0 signifies proper completion of the request. Along with the normal completion, the DPL return code field can provide a conditional completion code. All completion codes are documented in the section covering the DNR.

Completion

On successful completion, dirsrv() returns a 0. Otherwise it returns the general completion code of the DIRSRV macro of the DNR subsystem. The user should refer to the documentation on the DIRSRV macro for more information on error codes it can return.

Directory Services Parameter List (DPL)

This section includes the assembler language and C language definition for the Directory Services Parameter List (DPL). By convention, the assembler language definition refers to the Directory Services Parameter List as DPL; the C language definition refers to it as dpl.

Assembler Language Definition

This section contains the Assembler Language Definition for DPL, the Directory Services Parameter List.

DPL	DSECT	DIRECTORY SERVICES PARAMETER LIST
DPLIDENT DS	X	CONTROL BLOCK ID
DPLIDSTD EQU	237	STANDARD FORMAT ID
DPLFNCCD DS	X	FUNCTION CODE
DFHSTBYN EQU	1	GET HOST BY NAME
DFHSTBYV EQU	2	GET HOST BY VALUE
DFHSTBYA EQU	3	GET HOST BY ALIAS
DFNETBYN EQU	4	GET NETWORK BY NAME
DFNETBYV EQU	5	GET NETWORK BY VALUE
DFSRVBYN EQU	6	GET SERVICE BY NAME
DFSRVBYV EQU	7	GET SERVICE BY VALUE
DFPROBYN EQU	8	GET PROTOCOL BY NAME
DFPROBYV EQU	9	GET PROTOCOL BY VALUE
DFHSTSRV EQU	10	GET HOST SERVICES (BY NAME)
DFHSTINF EQU	11	GET HOST INFORMATION (BY NAME)
DFROUTE EQU	12	GET MAIL ROUTE
DFRPCBYN EQU	13	GET RPC BY NAME
DFRPCBYV EQU	14	GET RPC BY VALUE
DFMAX EQU	DFRPCBYV	MAXIMUM FUNCTION CODE
DPLACTIV DS	X	SEMAPHORE (DPL ACTIVE)
DPLFLAGS DS	X	FLAG BYTE
DPLFCMPL EQU	B'10000000'	DPL COMPLETED
DPLFCERR EQU	B'01000000'	COMPLETED WITH ERROR
DPLFXECB EQU	B'00100000'	DPLECBXR IS EXTERNAL ECB
DPLFEXIT EQU	B'00010000'	DPLECBXR IS EXIT ROUTINE
DPLF31B EQU	B'00001000'	REQUEST ISSUED WITH AMODE=31
* EQU	B'00000100'	RESERVED
* EQU	B'00000010'	RESERVED
* EQU	B'00000001'	RESERVED
DPLSYSID DS	CL4	MVS SUBSYSTEM ID
DPLECBXR DS	A	IECB/XECB/EXIT
ORG	*-4	
DPLECB DS	0F	ECB PARAMETER
DPLIECB DS	0F	INTERNAL ECB
DPLXECB DS	0A	EXTERNAL ECB ADDRESS
DPLEXIT DS	0A	EXIT ROUTINE ADDRESS
ORG	*+4	
EJECT		
DPLOPTCD DS	F	OPTION CODES
ORG	DPLOPTCD	
DPLOPCD1 DS	X	OPTION CODE #1
DOASYN EQU	B'10000000'	OPTCD=SYNC ASYN
DONOCOPY EQU	B'01000000'	OPTCD=COPY ORIGINAL
DOLOCAL EQU	B'00100000'	OPTCD=GLOBAL LOCAL
DONOBLOK EQU	B'00010000'	OPTCD=BLOCK NOBLOCK
* EQU	B'00001000'	RESERVED
* EQU	B'00000100'	RESERVED
* EQU	B'00000010'	RESERVED
* EQU	B'00000001'	RESERVED
DPLOPCD2 DS	X	OPTION CODE #2
DPLOPCD3 DS	X	OPTION CODE #3
DPLOPCD4 DS	X	OPTION CODE #4

Directory Services Parameter List (DPL)

DPLRTNCD	DS	F	COMPOSITE RETURN CODE
	ORG	DPLRTNCD	
DPLACTCD	DS	X	RECOVERY ACTION CODE
DAOKAY	EQU	0	SUCCESSFUL COMPLETION
DAEXCPTN	EQU	4	EXCEPTIONAL CONDITION
DAENVIRO	EQU	8	ENVIRONMENTAL CONDITION
DAFORMAT	EQU	12	FORMAT OR SPECIF ERRORS
DAPROCED	EQU	16	SEQUENCE AND PROCED ERRORS
DADPLERR	EQU	20	LOGIC ERRORS W/NO DPL RTNCD
DPLERRCD	DS	X	SPECIFIC ERROR CODE
DCOKAY	EQU	B'00000000'	00: NO CONDITIONALS
DCMORE	EQU	B'10000000'	00: MORE OCCUR. THAN SIZE
DCALIAS	EQU	B'01000000'	00: NAME WAS AN ALIAS
DCOVRFLO	EQU	B'00100000'	00: QNBUF TOO SMALL
DCNAMEIA	EQU	B'00010000'	00: NAME WAS INT ADDRESS
DCLOCAL	EQU	B'00001000'	00: GLOBAL USED LOCAL
DENONAME	EQU	1	04: NAME BUFFER/LEN ZERO
DENOVALU	EQU	2	04: VALUE BUFFER/LEN ZERO
DENOQNAM	EQU	3	04: QUAL NAME LEN ZERO
DETIMOUT	EQU	4	04: REQUEST TIMED OUT
DERFAIL	EQU	5	04: RESOLVER PROCESS FAILED
DENOTFND	EQU	6	04: REQUEST NOT FOUND
DENOCDS	EQU	7	04: NO CONFIG DATASET
DENAMERR	EQU	8	04: NAME DOES NOT EXIST
DEOVRFLO	EQU	9	04: RESULT BUFFER TOO SMALL
DENOBLOK	EQU	10	04: NOT READILY AVAILABLE
DENODATA	EQU	11	04: SERVER HAS NO DATA
DENAMODE	EQU	12	04: 31B NABUF/AMODE 24
DEVAMODE	EQU	13	04: 31B VABUF/AMODE 24
DEQNMODE	EQU	14	04: 31B QNBUF/AMODE 24
DESYSERR	EQU	1	08: SYSTEM ERROR
DESUBSYS	EQU	2	08: SUBSYSTEM ERROR
DENOTCNF	EQU	3	08: SUBSYS NOT INSTALLED
DENOTACT	EQU	4	08: SUBSYS NOT ACTIVE
DENOTRDY	EQU	5	08: SUBSYS NOT INITIALIZED
DESTOP	EQU	6	08: SUBSYS IS STOPPING
DEUNAVBL	EQU	7	08: UNAVAIL SERV/FACILITY
DERSOURC	EQU	8	08: INSUFFICIENT RESOURCES
DENOTPRB	EQU	9	08: USER RB NOT PRB
DETERM	EQU	10	08: SUBSYS HAS TERMINATED
DEBDOPCD	EQU	1	12: INVALID OPTION CODE
DEBDFNCD	EQU	2	12: INVALID FUNCTION CODE
DEBDXECB	EQU	3	12: INVALID ECB ADDRESS
DEBDEXIT	EQU	4	12: INVALID EXIT ADDRESS
DEBDNAME	EQU	5	12: INVALID DOMAIN NAME
DEBDVALU	EQU	6	12: INVALID VALUE
DEBDQNAM	EQU	7	12: INVALID QUAL NAME BUFF
DEACTIVE	EQU	1	16: DPL IS STILL ACTIVE
DEBDTYPE	EQU	1	20: DPL STNDRD FORM NOT 237
DEPROTCT	EQU	2	20: DPL IS FETCH/STORE PROT
DEPLMODE	EQU	3	20: 31B DPL PTR/AMODE 24
DPLDGNCD	DS	H	DIAGNOSTIC AND SENSE CODES
DPLFXPAR	DS	XL(12) 4X3	FIXED-LENGTH (INTERNAL) PARMS
	ORG	DPLFXPAR	
DPLTIME	DS	F	TIME LIMIT
DPLSIZE	DS	F	SIZE LIMIT
	DS	F	RESERVED
DPLVAPAR	DS	XL(24) 8X3	VARIABLE-LENGTH (EXTERNAL) PARMS
	ORG	DPLVAPAR	
DPLNAME	DS	XL(8) 4X2	PURPORTED NAME
	ORG	DPLNAME	
DPLNABUF	DS	A	PARAMETER ADDRESS
DPLNALEN	DS	F	PARAMETER LENGTH
DPLVALUE	DS	XL(8) 4X2	VALUE PARAMETER
	ORG	DPLVALUE	

```

DPLVABUF DS      A                      PARAMETER ADDRESS
DPLVALEN DS      F                      PARAMETER LENGTH
DPLQNAME DS      XL(8)    4X2          FULLY QUALIFIED NAME
                ORG    DPLQNAME
DPLQNBUFF DS      A                      PARAMETER ADDRESS
DPLQNLEN DS      F                      PARAMETER LENGTH
DPLLEN    EQU     *-DPL                  LENGTH OF DPL
*   GENERAL RETURN CODES (RETURNED IN R15) ARE USED TO INDICATE
*   SUCCESSFUL OR UNSUCCESSFUL COMPLETION OF A REQUEST IN
*   SYNCHRONOUS MODE, AND ACCEPTANCE OR NON-ACCEPTANCE OF A
*   REQUEST IN ASYNCHRONOUS MODE.

DROKAY    EQU     0          SUCCESSFUL COMPLETION/ACCEPTED
DRFAILED  EQU     4          UNSUCCESS COMPLETION/NOT ACCEPT DRFATLPL
                EQU     8          FATAL DPL ERROR

```

C Language Definition

This section contains the C Language Definition for dpl, the Directory Services Parameter List.

```
#define DS_H

/*
 * exit address or event control block union
 */
union dplecbexit
{
    union
    {
        unsigned long    dpliecb;    /* internal ECB */
        unsigned char    *dplxecb;   /* external ECB address */
    } dplecb;
    void                (*dplexit)(); /* address of exit routine */
} ;

/*
 * option codes
 */

union dploptcd
{
    unsigned long        dploptcdl;   /* options as long */
    struct
    {
        unsigned char    dplopcd1;    /* options as bytes */
        unsigned char    dplopcd2;    /* options as bytes */
        unsigned char    dplopcd3;    /* options as bytes */
        unsigned char    dplopcd4;    /* options as bytes */
    } dploptcds;
} ;

/*
 * return code field union
 */

union dplrtncd
{
    unsigned long        dplrtncdl;    /* return completion stat*/
    struct
    {
        unsigned char    dplactcd;    /*recovery action code */
        unsigned char    dplerrcd;    /*specific error code */
        unsigned short    dpldgncd;    /*diagnostic sense codes */
    } dplrtncds;
} ;

/*
 * fixed-length parameters for dpl
 */

struct dplfxpar
{
    unsigned long        dpltime;      /* time limit */
    unsigned long        dplsize;      /* size limit */
    unsigned long        dplfxrsvd;    /* reserved */
};
```

```

/*
 * variable-length parameters for purported name
 */

struct dplname
{
    char            *dplnabuf;        /* parameter address */
    unsigned long    dplnalen;        /* parameter length */
};

/*
 * variable-length parameters for value parameters
 */

struct dplvalue
{
    char            *dplvabuf;        /* parameter address */
    unsigned long    dplvalen;        /* parameter length */
};

/*
 * variable-length parameters for fully qualified name
 */

struct dplqname
{
    char            *dplqnbuf;        /* parameter address */
    unsigned long    dplqnlen;        /* parameter length */
};

/*
 * variable-length parameters
 */

struct dplvapar
{
    struct dplname    dplname;        /* purported name */
    struct dplvalue    dplvalue;      /* value parameter*/
    struct dplqname    dplqname;      /* fully qualif. name */
};

/*
 * DNR request structure for calling the DIRSRV() function.
 */

struct dpl
{
    unsigned char    dplident;        /* control block ID */
    unsigned char    dplfnccd;        /* func code of req */
    unsigned char    dplactiv;        /* activity semaphore */
    unsigned char    dplflags;        /* flags used by API */
    unsigned char    dplsystid[4];    /* MVS subsystem ID */
    union dplecbexit dplecbexit;      /* ecb/exit rtn addr */
    union dploptcd   dploptcd;        /* option codes */
    union dplrtncd   dplrtncd;        /* return codes */
    struct dplfxpar   dplfxpar;        /* fixed length parms */
    struct dplvapar   dplvapar;        /* variable length parms */
};

```

```

};
/*
 * dplident
 */#define DPLIDSTD 237
/*
 * dplfnccd
 */

#define DFHSTBYN 1 /* get host by name */
#define DFHSTBYV 2 /* get host by value */
#define DFHSTBYA 3 /* get host by alias */
#define DFNETBYN 4 /* get network by name */
#define DFNETBYV 5 /* get network by value */
#define DFSRVBYN 6 /* get service by name */
#define DFSRVBYV 7 /* get service by value */
#define DFPROBYN 8 /* get protocol by name */
#define DFPROBYV 9 /* get protocol by value */
#define DFHSTSRV 10 /* get services by name */
#define DFHSTINF 11 /* get host info by name */
#define DFROUTE 12 /* get mail route */
#define DFRPCBYN 13 /* get rpc by name */
#define DFRPCBYV 14 /* get rpc by value */
#define DFMAX DFRPCBYV /* MAXIMUM function code */

/*
 * dplflags
 */

#define DPLFCMPL 0x80 /* DPL completed */
#define DPLFCERR 0x40 /* Completed with error */
#define DPLFECB 0x20 /* external ECB in use */
#define DPLFEXIT 0x10 /* exit in use */
#define DPLF31B 0x08 /* req iss in AMODE=31 */

/*
 * dplopctl
 */

#define DOASYNC 0x80 /* OPTCD=SYNC | ASYNC */
#define DONOCOPY 0x40 /* OPTCD=COPY | ORIGINAL */
#define DOLOCAL 0x20 /* OPTCD=GLOBAL | LOCAL */
#define DONOBLOCK 0x10 /* OPTCD=BLOCK | NOBLOCK */

/*
 * dplrtncd.dplactcd
 */

#define DAOKAY 0 /* successful completion */
#define DAEXCPTN 4 /* exceptional condition */
#define DAENVIRO 8 /* environmental cond */
#define DAFORMAT 12 /* format/specif error */
#define DAPROCED 16 /* sequence/proced error */
#define DADPLERR 20 /* logic error (no rtn code) */
/*
 * dplrtncd.dplerrcd
 */

```

```

#define DCOKAY          0x00      /* 00: no conditionals */
#define DCMORE          0x80      /* 00: more occur thn siz */
#define DCALIAS         0x40      /* 00: name was alias */
#define DCOVRFLO        0x20      /* 00: QNBUF too small */
#define DCNAMEIA        0x10      /* 00: name was IA */
#define DCLOCAL         0x08      /* 00: global used local */

#define DENONAME        1         /* 04: name buffer or len = 0 */
#define DENOVALU        2         /* 04: val buffer or len = 0 */
#define DENOQNAM        3         /* 04: qual name len = 0 */
#define DETIMOUT        4         /* 04: request timed out */
#define DERFAIL         5         /* 04: resolver failure*/
#define DENOTFND        6         /* 04: request not found */
#define DENOCDS         7         /* 04: no config data set */
#define DENAMERR        8         /* 04: name doesn't exist */
#define DEOVRFLO        9         /* 04: result buff too small */
#define DENOBLOK        10        /* 04: not readily avail */
#define DENODATA        11        /* 04: server has no data */
#define DENAMODE        12        /* 04: 31B nabuf/amode 24 */
#define DEVAMODE        13        /* 04: 31B vabuf/amode 24 */
#define DEQNMODE        14        /* 04: 31B qnbuf/amode 24 */

#define DESYSERR        1         /* 08: system error */
#define DESUBSYS        2         /* 08: subsystem error */
#define DENOTCNF        3         /* 08: subsys not config */
#define DENOTACT        4         /* 08: subsys not active */
#define DENOTRDY        5         /* 08: subsys not initialized*/
#define DESTOP          6         /* 08: subsys is stopping */
#define DEUNAVBL        7         /* 08: unavai service/facil */
#define DERSOURCE        8         /* 08: insufficient resources */
#define DENOTPRB        9         /* 08: user RB not PRB */
#define DETERM          10        /* 08: subsystem has stopped */

#define DEBDOPCD        1         /* 12: invalid option code */
#define DEBDFNCD        2         /* 12: invalid function cd */
#define DEBDXECB        3         /* 12: invalid ecb address */
#define DEBDEXIT        4         /* 12: invalid exit address */
#define DEBDNAME        5         /* 12: invalid domain name */
#define DEBDVALU        6         /* 12: invalid value */
#define DEBDQNAM        7         /* 12: inval qualified name */

#define DEACTIVE        1         /* 16: DPL is active */

#define DEBDTPYE        1         /* 20: inval DPL identifier */
#define DEPROTCT        2         /* 20: fetch/store protect */
#define DEPLMOD          3         /* 20: 31B DPL/amode 24 */
/*
 * general return codes
 */

#define DROKAY          0         /* successful completion */
#define DRFAILED        4         /* unsuccess completion */
#define DRFATLPL        8         /* fatal DPL error */

```

