

# DNR Directory Services

---

This appendix describes DNR (Domain Name Resolver) directory services and how the DNR can be used to augment the use of API macro instructions. It includes these sections:

- Overview  
Describes the DNR directory service, its function, and its configuration.
- Directory Database  
Describes the Domain Name System (DNS) and local configuration data.
- Syntactic Rules for Names  
Describes the format and handling of locally managed names and simple domain names.
- Directory Services Calls  
Includes the calls used to access directory information services for the Domain Name Resolver (DNR).

## Overview

The DNR is a separate application program and can be configured to run in its own address space or in the same address space as the API. The function of DNR is to provide directory services to other application programs. These application programs generally use DNR to acquire network transport services for communicating with remote programs in a distributed processing environment. In fact, DNR uses the API to provide the transport between components of a distributed database that comprises a global network directory.

## The Domain Name System (DNS)

The directory consists of locally configured information and global information by supporting the Domain Name System (DNS). The DNS defines a distributed database system and provides a more dynamic information retrieval system than a local database search. Information may be obtained from the DNR by invoking the DIRSRV macro instruction or the `dirsrv()` C function. The client provides the name of an object known to the DNR. The DNR searches the directory for the information associated with that name or attribute and returns the information to the client.

Directory services can be used to build transport protocol addresses by mapping host names and service names into network addresses and TCP or UDP port numbers.

## Other Directory Services

Other directory services are available to support application programs that process and distribute electronic mail. These services are currently provided by DNR:

- Given host name, return network address
- Given alias name, return official host name
- Given network address, return official host name
- Given network name, return network number
- Given network number, return network name
- Given protocol name and service name, return associated transport protocol address (in other words, TCP or UDP port number)
- Given port number, return associated service name
- Given protocol name, return official protocol number
- Given official protocol number, return protocol name
- Given host name, return CPU and operating system information
- Given host name, return list of well-known services supported by host
- Given host name, return list of host names designated as mail routers for the given host

## Directory Database

The DNR information base consists of locally configured information and global information. Corresponding to this data, some services involve accessing local configured information only. Others use local and global distributed database, satisfied by DNR's implementation of the Domain Name System (DNS).

## Domain Name System

The Domain Name System (DNS) consists of the domain name space, resource records that describe the name space, name servers, and resolvers. The domain name space is a tree structured name space intended to provide a mechanism for naming resources in such a way that the host, network, or protocol is transparent. Each node within the name space corresponds to a set of resources. These resources are identified in a defined data format and referred as Resource Records (RR). Name servers and resolvers are application programs that exchange the Resource Records describing the name space.

The hierarchical structure of the domain name space is reflected in the format of a domain name. Each DNS node is represented by a label that is the simple name of the node. A fully qualified domain name describes a path through the domain name space to a particular node, beginning with the top-level (root) node. The name is formed by concatenating the simple names (or labels) of each node in right-to-left sequence, separated by periods (.) and beginning with the top-level domain. Thus, a 4-level domain name appears in this format:

```
level-4.level-3.level-2.level-1
```

---

Information about the name space is divided into zones and held by the domain's name servers. A name server's function is to hold information about the domain name space and provide answers to resolver's requests. The resolver's function is to extract information from name servers in response to client requests. The DNR implements the resolver portion of the DNS and queries name servers for responses.

## Local Configuration Data

Local configuration data enable DNR to provide services not provided by the Domain Name System as well as to provide an alternative to network based directory.

## Syntactic Rules for Names

Names provided to DNR by the application program must conform to certain syntactic rules. Also, any name returned to the application program by DNR conforms to the same rules. These rules are defined separately for locally-managed names and Internet domain names.

## Locally-Managed Names

Locally-managed names consist of those names defined in locally-maintained configuration data sets, and may be any one of these types:

- Alias name (for hosts only)
- Network name
- Service name
- Protocol name

Such names are case-insensitive and consist of alphanumeric characters from the EBCDIC character set (a-z, A-Z, 0-9). The dash character (-) may also be used as long as it is embedded within the name (in other words, does not appear at the beginning or end of the name). Locally-managed names must be less than or equal to 40 characters in length.

## Simple Domain Names

Simple domain names (in other words, domain name labels) are case-insensitive and consist of alphanumeric characters from the EBCDIC character set (a-z, A-Z, 0-9). The dash character (-) may also be used as long as it is embedded within the name (in other words, does not appear at the beginning or end of the name). Simple domain names must be less than or equal to 63 characters in length, and fully-qualified domain names must be less than or equal to 255 characters in length, including the terminating period.

The period at the end of a domain name represents the root of the Domain Name System (DNS) and indicates that the name is fully-qualified. A domain name not terminated with a period is assumed to be partially-qualified. In the latter case, DNR constructs fully-qualified names by appending qualifiers from a search list in a predetermined order.

## Directory Services Calls

This section includes the calls used to access directory information services for the Domain Name Resolver (DNR). Each macro instruction description includes this information:

- The name of the macro instruction
- A brief statement of its function and use
- The assembler format description
- A detailed description of each operand
- A description of completion information returned
- A table of return codes
- A general usage information

It is assumed that you are familiar with the API concepts and facilities presented in *Cisco IOS for S/390 Assembler API Concepts*.

---

## DIRSRV

The DIRSRV macro instruction is defined in terms of its general use in obtaining information from the directory. Specific instances of the macro instruction are defined which correspond to the specific directory services that were listed in Other Directory Services.

```
[symbol] DIRSRV function_code,  
             information_category,  
             search_argument,  
             NABUF = name_address,  
             NALEN = name_length,  
             VABUF = value_address,  
             VALEN = value_length  
             [,QNBUF = qualified_name_address]  
             [,QNLEN = qualified_name_length]  
             [,SYSID = MVS_subsystem_id]  
             [,TIME = time_limit]  
             [,SIZE = size_limit]  
             [,OPTCD = ([SYNC | ASYNC]  
                       [,BLOCK | NOBLOCK]  
                       [,LOCAL | GLOBAL]  
                       [,COPY | ORIGINAL]))]  
             [,ECB = INTERNAL | event_control_block_addr]  
             [,EXIT = exit_routine_address]  
             [,MF = (I | L | G | M | E,dpl_address)]
```

### Syntax Description

*function\_code*

These values can be coded:

GET - Indicates that the requested information is to be retrieved from the directory and returned to the application program.

PURGE - Indicates a previously issued asynchronous DIRSRV request is to be purged.

<i>information_category</i>	<p>Specifies a category of information maintained by DNR. This operand in combination with the search argument determines the specific type of information requested by the application program. These values may be coded:</p> <p>HOST - Indicates information regarding a specific host consisting of its official name, alias name, and network-layer addresses (in other words, Internet addresses).</p> <p>NETWORK - Indicates information regarding a specific network consisting of its official name and network number.</p> <p>SERVICE - Indicates information regarding a well-known service consisting of its official name and transport-layer address (in other words, TCP or UDP port number).</p> <p>PROTOCOL - Indicates information regarding a specific protocol consisting of its official name and protocol number.</p> <p>HOSTSERV - Indicates information regarding a specific host consisting of its network-layer addresses and associated well-known services available at those addresses.</p> <p>HOSTINFO - Indicates information regarding a specific host consisting of a standardized CPU and operating system name.</p> <p>ROUTE - Indicates information regarding electronic mail routes consisting of a list of host names willing to act as mail routers for a specific host.</p>
<i>search_argument</i>	<p>Determines how the directory database is searched by indicating the specific type of information provided by the application program and implying the type of information to be returned by DNR. These values may be coded for this operand:</p> <p>BYNAME - Indicates that a name has been supplied by the application program in the storage area defined by the NABUF and NALen operands, and that the value associated with the named object is to be returned in the storage area defined by the VABUF and VALEN operands.</p> <p>BYVALUE - Indicates that a value has been supplied by the application program in the storage area defined by the VABUF and VALEN operands, and that the name of the object with the specified value is to be returned in the storage area defined by the NABUF and NALen operands.</p> <p>BYALIAS - Indicates that an alias name (or real name) has been supplied by the application program in the storage area defined by the NABUF and NALen operands, and that the real name associated with the alias name is to be returned in the storage area defined by the VABUF and VALEN operands.</p>
<b>NABUF</b> = <i>name_address</i>	<p>Indicates the address of a storage area in which the application program has placed the name of a host, network, service, or protocol, or in which DNR returns such a name. The type of name contained or returned in the storage area is determined by the values indicated for information category and search argument, and consists of a case-insensitive string of EBCDIC characters. Other syntactic rules may apply depending on the type of name.</p> <p>Default: 0 (no name storage area)</p>

---

<b>NALEN</b> = <i>name_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the NABUF operand. If the storage area contains a name provided by the application program, NALEN is the actual length of the name. If the storage area is to receive a name returned by DNR, NALEN is the maximum length of the storage area. The length is updated when the request is completed to reflect the actual length of the name returned. If no information is returned, the request is completed abnormally and the length remains unchanged. A length of zero is invalid.</p> <p>Default: 0 (no name provided or returned)</p>
<b>VABUF</b> = <i>value_address</i>	<p>Indicates the address of a storage area in which the application program has placed an Internet address, network number, TCP or UDP port number, or transport protocol number, or in which DNR returns such a value. The type of value contained or returned in the storage area is determined by the values indicated for information category and search argument. The format of such values is type-dependent, and the length is indicated by the VALEN operand.</p> <p>Default: 0 (no value storage area)</p>
<b>VALEN</b> = <i>value_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the VABUF operand. If the storage area contains a value provided by the application program, VALEN is the actual length of the value. If the storage area is to receive a value returned by DNR, VALEN is the maximum length of the storage area. The length is updated when the request is completed to reflect the actual length of the value returned. If no information is returned, the request is completed abnormally and the length remains unchanged. A length of zero is invalid.</p> <p>Default: 0 (no value provided or returned)</p>
<b>QNBUF</b> = <i>qualified_name_address</i>	<p>Indicates the address of a storage area in which DNR returns the fully-qualified name associated with a partial name provided by the application program. The partially-qualified name is contained in the storage area indicated by the NABUF and NALEN operands. Rules governing the formation of names and local configuration information maintained by DNR is used to construct fully-qualified names, which in turn are used to search the directory for the desired information.</p> <p>The value returned depends on the type of search being performed. If the search argument is BYALIAS, the name returned is the locally-qualified alias that is used to search for the real name. Otherwise, the search argument must be BYNAME and the name returned is the locally-qualified name, or if an alias is encountered in the search, the last de-referenced alias name.</p> <p>Default: 0 (no qualified name storage area)</p>
<b>QNLEN</b> = <i>qualified_name_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully-qualified name returned. A length of zero indicates that no name is to be returned. If QNLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.</p> <p>Default: 0 (no qualified name returned)</p>

<b>SYSID</b> = <i>MVS_subsystem_id</i>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)
<b>TIME</b> = <i>time_limit</i>	Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, an error code indicating such is returned to the application program.  Default: 0 (use MAXTIME limit in DNRCFG00)
<b>SIZE</b> = <i>size_limit</i>	Indicates a limit on the amount of information returned. The ultimate limit is determined by the length of the storage area provided to receive the returned values. However, for those requests that may return lists or an array of values, this operand limits the number of elements returned. A value of zero indicates there is no limit (other than the size of the storage area provided).  Default: 0 (no size limit)
<b>OPTCD</b> = SYNC   ASYNC	Indicates the synchronization mode to be used when executing this macro instruction.  Of the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.  If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.  Default: SYNC (synchronous mode)



---

**OPTCD = BLOCK |  
NOBLOCK**

Indicates the disposition of a request if the information is not immediately available.

If the option is OPTCD=BLOCK, processing continues in accordance with the synchronization mode selected (OPTCD=SYNC | ASYNC). This option code affects disposition of a request after it has been accepted by DNR.

If the option is OPTCD=NOBLOCK, the request is completed abnormally and the specific error code is set to DENOBLK indicating that the request could not be completed immediately. OPTCD=NOBLOCK is generally used in conjunction with OPTCD=SYNC to prevent suspension of the issuing task for an extended period of time.

Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (OPTCD=ASYNC).

Default: BLOCK (suspend task indefinitely)

**OPTCD=LOCAL |  
GLOBAL**

Indicates the scope of the directory search.

If the option is OPTCD=LOCAL, only the locally-maintained database is searched.

If the option is OPTCD=GLOBAL, the distributed database is searched, and queries for the requested information are sent to remote destinations as required.

If this option is specified, and access to the network is disabled or inoperative, or no distributed nodes are configured, the request proceeds as if OPTCD=LOCAL were specified.

Default: GLOBAL (global scope)

**OPTCD = COPY |  
ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is OPTCD=COPY, a local copy of the requested information may be used to satisfy the request. Specifying OPTCD=COPY lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations. If the option is OPTCD=ORIGINAL, the information must be obtained from its original, authoritative source. Specifying OPTCD=ORIGINAL assures that the most current information is returned.

Default: COPY (use copy of original data)

**ECB = INTERNAL |**  
*event\_control\_block\_addr* Indicates the location of an Event Control Block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in EXIT=exit routine address below.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT =**  
*exit\_routine\_address* Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF = (I | L | G | M | E,**  
*[dpl\_address]*) Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALLEN operands is not the fully qualified name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the fully qualified name in the QNBUF storage area. If the fully qualified name could not fit in the storage area, DCOVERFLO is set. If the application program specified a SIZE limit less than the defined number of entries for the given host, or if the entire list of return information would not fit in the storage area provided, the DCMORE conditional completion code is returned.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

If the fully qualified search string did not conform to the syntactic rules described in Syntactic Rules for Names, a DEBDNAME error code is returned. If the fully qualified search string is a valid host name but the host does not exist, the DNR returns an error code of DENAMERR. If the fully qualified search string is a valid and existing host but there is no specific data configured to satisfy the request, the DNR returns an error code of DENODATA. The fully qualified search string is the result of a fully qualified name given in the NABUF storage area, a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.

## Return Codes

This table lists the symbolic return codes for the DIRSRV macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-1 DIRSRV Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY DCOVERFLO	DCMORE DCNAMEIA	DCALIAS DCLOCAL
DRFAILED	DAEXCPTN	DENONAME DETIMOUT DENOCDS DENOBLK DEVAMODE	DENOVALU DERFAIL DENAMERR DENODATA DEQNMODE	DENOQNAM DENOTFND DEOVRFLO DENAMODE
		DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DENOTCNF DESTOP DENOTPRB
		DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

## GET-HOST-BYNAME

The GET-HOST-BYNAME instance of the DIRSRV macro instruction is used to obtain a list of Internet addresses for a given host when its name is known. The name provided may be a local alias or a partial or fully qualified domain name, and the information returned may be obtained locally from the host-name configuration data set (DNRHSTxx) or globally from the Internet Domain Name System (DNS).

```
[symbol] DIRSRV GET,HOST,BYNAME,
      NABUF = name_address,|
      NALEN = name_length,
      VABUF = value_address,
      VALEN = value_length
      [,QNBUF = qualified_name_address]
      [,QNLN = qualified_name_length]
      [,SYSID = MVS_subsystem_id]
      [,TIME = time_limit]
      [,SIZE = size_limit]
      [,OPTCD = ((SYNC | ASYNC)
                [BLOCK | NOBLOCK]
                [LOCAL | GLOBAL]
                [COPY | ORIGINAL])]
      [,ECB = INTERNAL | event_control_block_addr]
      [,EXIT = exit_routine_address]
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,HOST,BYNAME</b>	Indicates that the DNR is to search for and return a list of Internet addresses associated with the host name or alias provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the name of a host. The name may be a local alias defined in the alias configuration member (DNRALCxx), a local host name defined in the host configuration member (DNRHSTxx), or an Internet domain name or alias name defined by the DNS. Names may be partially or fully qualified. All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a fully qualified domain name is 255 bytes. The maximum length of a local alias name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)

---

<b>VABUF</b> = <i>value_address</i>	<p>Indicates the address of a storage area in which DNR returns a list of Internet addresses associated with the named host. Each Internet address is four bytes. The number of addresses returned is the smaller of the total number defined for the host, the length of the storage area divided by the length of an Internet address (4), or the number indicated by the SIZE operand.</p> <p>Default: 0 (no value storage area)</p>
<b>VALEN</b> = <i>value_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged. The minimum length of the storage area is 4 bytes. It is not necessary that the storage area length be evenly divisible by the length of an Internet address.</p> <p>Default: 0 (no value returned)</p>
<b>QNBUF</b> = <i>qualified_name_address</i>	<p>Indicates the address of a storage area in which DNR returns the fully qualified name used to search for the requested information. The fully qualified name is either the result of a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.</p> <p>Default: 0 (no qualified name storage area)</p>
<b>QNLEN</b> = <i>qualified_name_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully qualified name returned. If QNLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.</p> <p>Default: 0 (no qualified name returned)</p>
<b>SYSID</b> = <i>MVS_subsystem_id</i>	<p>Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.</p> <p>Default: not indicated (use installation default)</p>

**TIME** = *time\_limit*

Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, DETIMOUT error code is returned to the application program. TIME is only applicable if OPTCD=GLOBAL.

Default: 0 (use MAXTIME limit in DNRCFG00)

**SIZE** = *size\_limit*

Indicates a limit on the amount of information returned for requests that return lists or arrays of values. A value of zero indicates there is no limit and the DNR is to return all Internet addresses associated with the domain name. If the return information would not fit in its entirety in the storage area provided or if the application program specifies a limit less than the defined number of Internet addresses for the given host, the DCMORE conditional completion code is returned.

Default: 0 (no size limit)

**OPTCD** = SYNC | ASYNC

Indicates the synchronization mode to be used when executing this macro instruction.

If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.

If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.

Default: SYNC (synchronous mode)

---

**OPTCD = BLOCK | NOBLOCK**

Indicates the disposition of a request if the information is not immediately available.

If the option is OPTCD=BLOCK, processing continues in accordance with the synchronization mode selected (OPTCD=SYNC | ASYNC).

If the option is OPTCD=NOBLOCK. The request is completed abnormally and the specific error code is set to DENOBLK indicating that the request could not be completed immediately. OPTCD=NOBLOCK is generally used in conjunction with OPTCD=SYNC to prevent suspension of the issuing task for an extended period of time.

Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (OPTCD=ASYNC).

OPTCD=BLOCK|NOBLOCK is only applicable if OPTCD=GLOBAL. This option code affects disposition of a request after it has been accepted by the DNR.

Default: BLOCK (suspend task indefinitely)

**OPTCD = LOCAL | GLOBAL**

Indicates the scope of the directory search.

If the option is OPTCD=LOCAL, only the locally-maintained database is searched.

If the option is OPTCD=GLOBAL, the distributed database is searched, and queries for the requested information are sent to remote destinations as required.

If this option is specified, and access to the network is disabled or inoperative, or no distributed nodes are configured, the request proceeds as if OPTCD=LOCAL were specified.

Default: GLOBAL (global scope)

**OPTCD = COPY | ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is OPTCD=COPY, a local copy of the requested information may be used to satisfy the request. Specifying OPTCD=COPY lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations.

If the option is OPTCD=ORIGINAL, the information must be obtained from its original, authoritative source. Specifying OPTCD=ORIGINAL assures that the most current information is returned.

OPTCD=COPY|ORIGINAL is only applicable if OPTCD=GLOBAL.

Default: COPY (use copy of original data)

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
*dpl\_address*)

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)



## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALLEN operands are not the fully qualified name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the fully qualified name in the QNBUF storage area. If the fully qualified name could not fit in the storage area, DCOVERFLO is set. If the application program specified a SIZE limit less than the defined number of entries for the given host, or if the entire list of return information would not fit in the storage area provided, the DCMORE conditional completion code is returned.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

If the fully qualified search string did not conform to the syntactic rules described in Locally-Managed Names, a DEBDNAME error code is returned. If the fully qualified search string is a valid host name but the host does not exist, the DNR returns an error code of DENAMERR. If the fully qualified search string is a valid and existing host but there is no specific data configured to satisfy the request, the DNR returns an error code of DENODATA. The fully qualified search string is the result of a fully qualified name given in the NABUF storage area, a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.

## Return Codes

This table lists the symbolic return codes for the GET-HOST-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-2 GET-HOST-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY DCOVERFLO	DCMORE DCNAMEIA	DCALIAS DCLOCAL
DRFAILED	DAEXCPTN	DENONAME DETIMOUT DENOCDS DENOBLK DEVAMODE	DENOVALU DERFAIL DENAMERR DENODATA DEQNMODE	DENOQNAMDENO TFND DEOVRFLO DENAMODE
	DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DESUBSYS DENOTRDY DERSOURC	DENOTCNF DESTOP DENOTPRB
	DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME	DEBDXECB DEBDVALU

**Table 2-2 GET-HOST-BYNAME Return Codes (Continued)**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

**Usage Information**

The GET-HOST-BYNAME instance of the DIRSRV macro is used to get a list of internet addresses given an Internet domain name.

- If OPTCD=LOCAL, the DNR returns information received from the host configuration member (DNRHSTxx).
- If OPTCD=GLOBAL, the DNR returns information received from the Domain Name System Address (A) records.

The list of returned Internet addresses is sorted according to the networks given in the network preference configuration member (DNRNPCxx).

**Example**

This example demonstrates the use of GET-HOST-BYNAME. The request is to find the list of internet addresses for the host NS.NASA.GOV. The application supplies this DIRSRV information:

NABUF = (address of:)

N	S	.	N	A	S	A	.	G	O	V						
---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--

NALEN = 11 VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100 QNBUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

QNLEN = 100  
 SIZE = 0

This information is returned:

NABUF = (address of:)

N	S	.	N	A	S	A	.	G	O	V						
---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--

NALEN = 11  
 VABUF = (address of:)

0x80	0x66	0x10	0x0a	0xc0	0x34	0xc3	0x0a									
------	------	------	------	------	------	------	------	--	--	--	--	--	--	--	--	--

---

VALEN = 8  
QNBUF = (address of:)

N	S	.	N	A	S	A	.	G	O	V	.				
---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

QNLEN = 12  
SIZE = 2

Because the search string, 'NS.NASA.GOV' is partially qualified, the search list strings were appended to 'NS.NASA.GOV'. The root (.) is always included in the search list, therefore, the directory was searched for 'NS.NASA.GOV.'. The DNR received a DNS response for 'NS.NASA.GOV.' and returned the information to the application program.

## GET-HOST-BYVALUE

The GET-HOST-BYVALUE instance of the DIRSRV macro instruction is used to obtain a fully qualified host name for a host when its address is known. The address provided may be an address defined locally or globally known to the Internet Domain Name System (DNS) and the information returned may be obtained locally from the host name configuration member (DNRHSTxx) or globally.

```
[symbol] DIRSRV GET,HOST,BYVALUE,
      NABUF = name_address,
      NALEN = name_length,
      VABUF = value_address,
      VALEN = value_length
      [,SYSID = MVS_subsystem_id]
      [,TIME = time_limit]
      [,OPTCD = ([SYNC | ASync]
                [,BLOCK | NOBLOCK]
                [,LOCAL | GLOBAL]
                [,COPY | ORIGINAL]))]
      [,ECB = INTERNAL | event_control_block_addr]
      [,EXIT = exit_routine_address]
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,HOST,BYVALUE</b>	Indicates that the DNR is to search for and return a fully qualified host name associated with the host Internet address provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the DNR returns a fully qualified host name associated with the given Internet address.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the storage area identified by the NABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no name returned)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which the application program has placed a four-byte Internet address.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The value of the storage area must be 4 bytes.  Default: 0 (no value provided)

- 
- SYSID** = *MVS\_subsystem\_id* Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.
- Default: not indicated (use installation default)
- TIME** = *time\_limit* Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, DETIMOUT error code is returned the application program. TIME is only applicable if OPTCD=GLOBAL.
- Default: 0 (use MAXTIME limit in DNRCFG00)
- OPTCD** = SYNC | ASYNC Indicates the synchronization mode to be used when executing this macro instruction.
- If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.
- If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.
- Default: SYNC (synchronous mode)

**OPTCD = BLOCK | NOBLOCK**

Indicates the disposition of a request if the information is not immediately available.

If the option is OPTCD=BLOCK, processing continues in accordance with the synchronization mode selected (OPTCD=SYNC | ASYNC).

If the option is OPTCD=NOBLOCK, the request is completed abnormally and the specific error code is set to DENOBLK indicating that the request could not be completed immediately. OPTCD=NOBLOCK is generally used in conjunction with OPTCD=SYNC to prevent suspension of the issuing task for an extended period of time.

Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (OPTCD=ASYNC).

OPTCD=BLOCK|NOBLOCK is only applicable if OPTCD=GLOBAL. This option code affects disposition of a request after it has been accepted by DNR.

Default: BLOCK (suspend task indefinitely)

**OPTCD = LOCAL | GLOBAL**

Indicates the scope of the directory search.

If the option is OPTCD=LOCAL, only the locally-maintained database is searched.

If the option is OPTCD=GLOBAL, the distributed database is searched, and queries for the requested information are sent to remote destinations as required.

If this option is specified, and access to the network is disabled or inoperative, or no distributed nodes are configured, the request proceeds as if OPTCD=LOCAL were specified.

OPTCD=COPY|ORIGINAL is only applicable if OPTCD=GLOBAL.

Default: GLOBAL (global scope)

**OPTCD = COPY | ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is OPTCD=COPY, a local copy of the requested information may be used to satisfy the request. Specifying OPTCD=COPY lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations.

If the option is OPTCD=ORIGINAL, the information must be obtained from its original, authoritative source. Specifying OPTCD=ORIGINAL assures that the most current information is returned.

Default: COPY (use copy of original data)

---

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner.

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
*[dpl\_address]*)

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

### Return Codes

This table lists the symbolic return codes for the GET-HOST-BYVALUE macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-3 GET-HOST-BYVALUE Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY	DCLOCAL	
DRFAILED	DAEXCPTN	DENONAME	DENOVALU	DENOQNAM
		DETIMOUT	DERFAIL	DENOTCFND
		DENOCDS	DEOVRFLO	DENOBLOK
	DAENVIRO	DENODATA	DENAMODE	DEVAMODE
		DEQNMODE		
		DESYSERR	DESUBSYS	DENOTCNF
	DAFORMAT	DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DEVSOURC	DENOTPRB
		DETERM		
DRFATLPTL	DAPROCED	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
		DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

### Usage Information

The GET-HOST-BYVALUE instance of the DIRSRV macro is used to obtain a fully qualified domain name given an Internet address.

- If OPTCD=LOCAL, the DNR returns information found in the host configuration member (DNRHSTxx).
- If OPTCD=GLOBAL, the DNR returns information received from Domain Name System Pointer (PTR) records.

### Example

This example demonstrates the use of GET-HOST-BYVALUE. The request is to find the host name for the internet address 192.52.195.10.



---

The application supplies this DIRSRV information:

NABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 100

VABUF = (address of:)

0xc0	0x34	0xc3	0x0a														
------	------	------	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 4

This information is returned:

NABUF = (address of:)

N	S	.	N	A	S	A	.	G	O	V	.						
---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--

NALEN = 12

VABUF = (address of:)

0xc0	0x34	0xc3	0x0a														
------	------	------	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 4

## GET-HOST-BYALIAS

The GET-HOST-BYALIAS instance of the DIRSRV macro instruction is used to obtain a fully qualified domain name for a host when its alias is known. The name provided may be a local alias or a partial or fully qualified domain name alias, and the information returned may be obtained locally from the alias configuration member (DNRALCxx) or globally from the Internet Domain Name System (DNS).

```
[symbol]DIRSRV GET,HOST,BYALIAS,
      NABUF = name_address,
      NALEN = name_length,
      VABUF = value_address,
      VALEN = value_length
      [,QNBUF = qualified_name_address]
      [,QNLLEN = qualified_name_length]
      [,SYSID = MVS_subsystem_id]
      [,TIME = time_limit]
      [,OPTCD = ([SYNC | ASYNC]
                [,BLOCK | NOBLOCK]
                [,LOCAL | GLOBAL]
                [,COPY | ORIGINAL])]
      [,ECB = INTERNAL | event_control_block_addr]
      [,EXIT = exit_routine_address]
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,HOST,BYALIAS</b>	Indicates that the DNR is to search for and return the fully qualified name associated with the alias provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the alias name of a host. The name may be a local alias defined in the alias configuration member (DNRALCxx) or an Internet domain alias name defined by the DNS. Names may be partially or fully qualified. All names must conform to the syntactic rules described in Locally-Managed Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the alias located in the storage area identified by the NABUF operand. The maximum length of a fully qualified domain name is 255 bytes. The maximum length of a local alias name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which the DNR returns a fully qualified name associated with the given alias.  Default: 0 (no value storage area)

---

<b>VALEN</b> = <i>value_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.</p> <p>Default: 0 (no value returned)</p>
<b>QNBUF</b> = <i>qualified_name_address</i>	<p>Indicates the address of a storage area in which DNR returns the fully qualified name used to search for the requested information. The fully qualified name is either the result of a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.</p> <p>Default: 0 (no qualified name storage area)</p>
<b>QNLLEN</b> = <i>qualified_name_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully qualified name returned. If QNLLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.</p> <p>Default: 0 (no qualified name returned)</p>
<b>SYSID</b> = <i>MVS_subsystem_id</i>	<p>Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.</p> <p>Default: not indicated (use installation default)</p>
<b>TIME</b> = <i>time_limit</i>	<p>Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, DETIMOUT error code is returned the application program. TIME is only applicable if OPTCD=GLOBAL.</p> <p>Default: 0 (use MAXTIME limit in DNRCFG00)</p>

<b>OPTCD = SYNC   ASYNC</b>	<p>Indicates the synchronization mode to be used when executing this macro instruction.</p> <p>If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.</p> <p>If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.</p> <p>Default: SYNC (synchronous mode)</p>
<b>OPTCD = BLOCK   NOBLOCK</b>	<p>Indicates the disposition of a request if the information is not immediately available.</p> <p>If the option is OPTCD=BLOCK, processing continues in accordance with the synchronization mode selected (OPTCD=SYNC   ASYNC).</p> <p>If the option is OPTCD=NOBLOCK, the request is completed abnormally and the specific error code is set to DENOBLK indicating that the request could not be completed immediately. OPTCD=NOBLOCK is generally used in conjunction with OPTCD=SYNC to prevent suspension of the issuing task for an extended period of time.</p> <p>Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (OPTCD=ASYNC).</p> <p>OPTCD=BLOCK NOBLOCK is only applicable if OPTCD=GLOBAL. This option code affects disposition of a request after it has been accepted by the DNR.</p> <p>Default: BLOCK (suspend task indefinitely)</p>
<b>OPTCD = LOCAL   GLOBAL</b>	<p>Indicates the scope of the directory search.</p> <p>If the option is OPTCD=LOCAL, only the locally-maintained database is searched.</p> <p>If the option is OPTCD=GLOBAL, the distributed database is searched, and queries for the requested information are sent to remote destinations as required.</p> <p>If this option is specified, and access to the network is disabled or inoperative, or no distributed nodes are configured, the request proceeds as if OPTCD=LOCAL were specified.</p> <p>Default: GLOBAL (global scope)</p>

---

**OPTCD = COPY | ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is OPTCD=COPY, a local copy of the requested information may be used to satisfy the request. Specifying OPTCD=COPY lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations.

If the option is OPTCD=ORIGINAL, the information must be obtained from its original, authoritative source. Specifying OPTCD=ORIGINAL assures that the most current information is returned.

OPTCD=COPY|ORIGINAL is only applicable if OPTCD=GLOBAL.

Default: COPY (use copy of original data)

**ECB = INTERNAL | *event\_control\_block\_addr***

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT = *exit\_routine\_address***

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl address*, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALLEN operands is not the fully qualified name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the fully qualified name in the QNBUF storage area. If the fully qualified name could not fit in the storage area, DCOVERFLO is set.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

If the fully qualified search string did not conform to the syntactic rules described in Syntactic Rules for Names, a DEBDNAME error code is returned. If the fully qualified search string is a valid host name but the host does not exist, the DNR returns an error code of DENAMERR. If the fully qualified search string is a valid and existing host but there is no specific data configured to satisfy the request (in other words, the search string is not an alias but an actual host name) the DNR returns an error code of DENODATA. The fully qualified search string is the result of a fully qualified name given in the NABUF storage area, a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.

### Return Codes

This table lists the symbolic return codes for the GET-HOST-BYALIAS macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-4 GET-HOST-BYALIAS Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY DCLOCAL	DCALIAS	DCOVRFLO
DRFAILED	DAEXCPTN	DENONAME DETIMOUT DENOCD DENOBLK DEVAMODE	DENOVALU DERFAIL DENAMERR DENODATA DEQNMODE	DENOQNAME TFND DEOVRFLO DENAMODE
		DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DENOTCNF DESTOP DENOTPRB
DRFATLPTL	DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME	DEBDXECB DEBDVALU
	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

### Usage Information

The GET-HOST-BYALIAS instance of the DIRSRV macro is used to obtain a fully qualified domain name given an alias.

If OPTCD=LOCAL, the DNR returns information received from the alias configuration member (DNRALCxx).

If OPTCD=GLOBAL, the DNR returns information received from Domain Name System Canonical (CNAME) records.

### Example

This example demonstrates the use of GET-HOST-BYALIAS, OPTCD=LOCAL. The request is to find a fully qualified domain name for the subsystem name ACSS.

The alias configuration member (DNRALCxx) contains:

ACSS ZEUS.ACC.COM.

The application supplies this DIRSRV information:

NABUF = (address of:)

A	C	S	S																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 4

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

## GET-HOST-BYALIAS

---

This information is returned:

NABUF = (address of:)

A	C	S	S												
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 4

VABUF = (address of:)

Z	E	U	S	.	A	C	C	.	C	O	M	.			
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

VALEN = 13



---

## GET-NETWORK-BYNAME

The GET-NETWORK-BYNAME instance of the DIRSRV macro instruction is used to obtain a network number when its name is known. The information returned is obtained locally from the network configuration member (DNRNETxx).

```
[symbol] DIRSRV GET,NETWORK,BYVALUE,  
          NABUF = name_address,  
          NALEN = name_length,  
          VABUF = value_address,  
          VALEN = value_length  
          [,SYSID = MVS_subsystem_id]  
          [,OPTCD = SYNC | ASYNC]  
          [,ECB = INTERNAL | event_control_block_addr]  
          [,EXIT = exit_routine_address]  
          [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,NETWORK,BYNAME</b>	Indicates that DNR is to search for and return a network number associated with the network name provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the name of a network. All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a local network name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided or returned)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which DNR returns a network number associated with the named network.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned.  If the request completes normally, the length is updated to a value from one to three.  If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no value returned)

<p><b>SYSID</b> = <i>MVS_subsystem_id</i></p>	<p>Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.</p> <p>Default: not indicated (use installation default)</p>
<p><b>OPTCD</b> = SYNC   ASYNC</p>	<p>Indicates the synchronization mode to be used when executing this macro instruction.</p> <p>If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.</p> <p>If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.</p> <p>Default: SYNC (synchronous mode)</p>
<p><b>ECB</b> = INTERNAL   <i>event_control_block_addr</i></p>	<p>Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.</p> <p>The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:</p> <p>If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.</p> <p>If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.</p> <p>If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.</p> <p>Default: INTERNAL (internal ECB)</p>

**EXIT** = *exit\_routine\_address* Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl* address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

Note: It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

### Return Codes

This table lists the symbolic return codes for the GET-NETWORK-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-5 GET-NETWORK-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
		DROKAY	DAOKAY	DCOKAY
DRFAILED	DAEXCPTN	DENONAMEDENO	DENOVALU	DENOTFND
		CDS DEVAMODE	DEOVRFLO	DENAMODE

**Table 2-5 GET-NETWORK-BYNAME Return Codes (Continued)**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
	DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DESUBSYS DENOTRDY DERSOURC	DENOTCNF DESTOP DENOTPRB
	DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME	DEBDXECB DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

**Example**

This example demonstrates the use of GET-NETWORK-BYNAME. The request is to find the network number for the name ARPANET.

The application supplies this DIRSRV information:

NABUF = (address of:)

A	R	P	A	N	E	T													
---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 7

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

This information is returned:

NABUF = (address of:)

A	R	P	A	N	E	T													
---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 7

VABUF = (address of:)

10																			
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1

---

## GET-NETWORK-BYVALUE

The GET-NETWORK-BYVALUE instance of the DIRSRV macro instruction is used to obtain a network name when its value is known. The information returned is obtained locally from the network configuration member (DNRNETxx).

```
[symbol] DIRSRV GET,NETWORK,BYVALUE,  
          NABUF = name_address,  
          NALEN = name_length,  
          VABUF = value_address,  
          VALEN = value_length  
          [,SYSID = MVS_subsystem_id]  
          [,OPTCD = SYNC | ASYNC]  
          [,ECB = INTERNAL | event_control_block_addr]  
          [,EXIT = exit_routine_address]  
          [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

**GET,NETWORK,BYVALUE** Indicates that DNR is to search for and return a network name associated with the network number provided by the application program.

NABUF = *name\_address*

Indicates the address of a storage area in which the DNR returns the name of a network.

Default: 0 (no name storage area)

**NALEN = *name\_length***

Indicates the length (in bytes) of the storage area identified by the NABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.

Default: 0 (no value returned)

**VABUF = *value\_address***

Indicates the address of a storage area in which the application program has placed a network number.

Default: 0 (no value storage area)

**VALEN = *value\_length***

Indicates the length (in bytes) of the network number located in the storage area identified by the VABUF operand. This number should be one to three (bytes). A length of zero is invalid.

Default: 0 (no value returned)

**SYSID = *MVS\_subsystem\_id***

Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.

Default: not indicated (use installation default)

**OPTCD = SYNC | ASYNC**

Indicates the synchronization mode to be used when executing this macro instruction.

If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.

If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.

Default: SYNC (synchronous mode)

**ECB = INTERNAL |**  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT = *exit\_routine\_address***

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl address*, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

### Return Codes

This table lists the symbolic return codes for the GET-NETWORK-BYVALUE macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-6 GET-NETWORK-BYVALUE Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY		
DRFAILED	DAEXCPTN	DENONAME	DENOVALU	DENOTFND
		DENOCDS	DEOVRFLO	DENAMODE
DRFATLPTL	DAENVIRO	DEVSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
		DETERM		
DRFATLPTL	DAFORMAT	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
		DADPLERR	DEBDTYPE	DEPROTCT

**Example**

This example demonstrates the use of GET-NETWORK-BYVALUE. The request is to find the network name for 10.

The application supplies this DIRSRV information:

NABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 100

VABUF = (address of:)

10																			
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1

This information is returned:

NABUF = (address of:)

A	R	P	A	N	E	T													
---	---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 7

VABUF = (address of:)

10																			
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1



---

## GET-PROTOCOL-BYNAME

The GET-PROTOCOL-BYNAME instance of the DIRSRV macro instruction is used to obtain a protocol number when its name is known. The information returned is obtained locally from the protocol configuration member (DNRPRTxx).

```
[symbol] DIRSRV GET,PROTOCOL,BYNAME,  
          NABUF = name_address,  
          NALEN = name_length,  
          VABUF = value_address,  
          VALEN = value_length  
          [,SYSID = MVS_subsystem_id]  
          [,OPTCD = SYNC | ASYNC]  
          [,ECB = INTERNAL | event_control_block_addr]  
          [,EXIT = exit_routine_address]  
          [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,PROTOCOL,BYNAME</b>	Indicates that the DNR is to search for and return a protocol number associated with the protocol name provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed a protocol name. All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a protocol name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which DNR returns a protocol number associated with the named protocol.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned.  If the request completes normally, the length is updated to a value from one to three.  If the request complete abnormally, no information is returned and the length remains unchanged.  Default: 0 (no value returned)

<p><b>SYSID</b> = <i>MVS_subsystem_id</i></p>	<p>Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.</p> <p>Default: not indicated (use installation default)</p>
<p><b>OPTCD</b> = SYNC   ASYNC</p>	<p>Indicates the synchronization mode to be used when executing this macro instruction.</p> <p>If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.</p> <p>If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.</p> <p>Default: SYNC (synchronous mode)</p>
<p><b>ECB</b> = INTERNAL   <i>event_control_block_addr</i></p>	<p>Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.</p> <p>The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:</p> <p>If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.</p> <p>If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.</p> <p>If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.</p> <p>Default: INTERNAL (internal ECB)</p>

**EXIT**=*exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl* address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

## Return Codes

This table lists the symbolic return codes for the GET-PROTOCOL-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-7 GET-PROTOCOL-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY		
DRFAILED	DAEXCPTN	DENONAMEDEN OCDS DEVAMODE	DENOVALU DEOVRFLO	DENOTFND DENAMODE

**Table 2-7 GET-PROTOCOL-BYNAME Return Codes (Continued)**

General Return Code	Recovery	Conditional Or Specific Error Code		
	Action Code			
	DAENVIRO	DESYSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
		DETERM		
	DAFORMAT	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

**Example**

This example demonstrates the use of GET-PROTOCOL-BYNAME. The request is to find the protocol number of TCP.

The application supplies this DIRSRV information:

NABUF = (address of:)

T	C	P																	
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 3

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

This information is returned:

NABUF = (address of:)

T	C	P																	
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 3

VABUF = (address of:)

6																			
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1

---

## GET-PROTOCOL-BYVALUE

The GET-PROTOCOL-BYVALUE instance of the DIRSRV macro instruction is used to obtain a protocol name when its official protocol number is known. The information returned is obtained locally from the protocol configuration member (DNRPRTxx).

```
[symbol] DIRSRV GET,PROTOCOL,BYVALUE,  
          NABUF = name_address,  
          NALEN = name_length,  
          VABUF = value_address,  
          VALEN = value_length  
          [,SYSID = MVS_subsystem_id]  
          [,SIZE = size_limit]  
          [,ECB = INTERNAL | event_control_block_addr]  
          [,EXIT = exit_routine_address]  
          [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,PROTOCOL,BYVALUE</b>	Indicates that DNR is to search for and return a protocol name associated with the protocol number provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the DNR returns a protocol name associated with the protocol number contained in the storage area indicated by the VABUF and VALEN operands.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the storage area identified by the NABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged. A length of zero is invalid.  Default: 0 (no name provided or returned)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which the application program has placed a protocol number.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the protocol number located in the storage area identified by the VABUF operand. This number should be 1 (byte). A length of zero is invalid.  Default: 0 (no value returned)
<b>SYSID = <i>MVS_subsystem_id</i></b>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)

**SIZE** = *size\_limit*

Indicates a limit on the amount of information returned. The ultimate limit is determined by the length of the storage area provided to receive the returned values. However, for those requests that may return lists or an array of values, this operand limits the number of elements returned. A value of zero indicates there is no limit (other than the size of the storage area provided).

Default: 0 (no size limit)

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in EXIT = *exit\_routine\_address*.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl address*, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

### Return Codes

This table lists the symbolic return codes for the GET-PROTOCOL-BYVALUE macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-8 GET-PROTOCOL-BYVALUE Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY		
DRFAILED	DAEXCPTN	DENONAME	DENOVALU	DENOTFND
		DENOCDS	DEOVRFLO	DENAMODE
DRFATLPTL	DAENVIRO	DEVSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
		DETERM		
DRFATLPTL	DAFORMAT	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
		DADPLERR	DEBDTYPE	DEPROTCT

**Example**

This example demonstrates the use of GET-PROTOCOL-BYVALUE. The request is to find a protocol name for the protocol number 6. The application supplies this DIRSRV information:

NABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 100

VABUF = (address of:)

6																			
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1

This information is returned:

NABUF = (address of:)

T	C	P																	
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 3

VABUF = (address of:)

6																			
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1



---

## GET-SERVICE-BYNAME

The GET-SERVICE-BYNAME instance of the DIRSRV macro instruction is used to obtain a port number when its name is known. The information returned is obtained locally from the services configuration member (DNRSVCxx).

```
[symbol] DIRSRV GET,SERVICE,BYNAME,  
          NABUF = name_address,  
          NALEN = name_length,  
          VABUF = value_address,  
          VALEN = value_length  
          [,SYSID = MVS_subsystem_id]  
          [,ECB = INTERNAL | event_control_block_addr]  
          [,EXIT = exit_routine_address]  
          [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,SERVICE,BYNAME</b>	Indicates that DNR is to search for and return a service port number associated with the protocol/service pair provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the protocol/service name pairs. The protocol/service names should be separated by a slash (/). All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a local protocol or service name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided or returned)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which DNR returns a service number associated with the named protocol/service pair.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned.  If the request completes normally, the length is updated to a value from one to three.  If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no value returned)

**SYSID** = *MVS\_subsystem\_id* Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.

Default: not indicated (use installation default)

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNCR), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in EXIT = exit\_routine\_address.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNCR), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNCR. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl address*, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

### Return Codes

This table lists the symbolic return codes for the GET-SERVICE-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-9 GET-SERVICE-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY		
DRFAILED	DAEXCPTN	DENONAME	DENOVALU	DENOTFND
		DENOCDS	DEOVRFLO	DENAMODE
DRFATLPTL	DAENVIRO	DEVSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
		DETERM		
DRFATLPTL	DAFORMAT	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
		DADPLERR	DEBDTYPE	DEPROTCT

**Example**

This example demonstrates the use of GET-SERVICE-BYNAME. The request is to obtain a service number for the name TCP/DISCARD.

The application supplies this DIRSRV information:

NABUF = (address of:)

T	C	P	/	D	I	S	C	A	R	D					
---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--

NALEN = 11

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

This information is returned:

NABUF = (address of:)

T	C	P	/	D	I	S	C	A	R	D					
---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--

NALEN = 11

VABUF = (address of:)

9															
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1

---

## GET-SERVICE-BYVALUE

The GET-SERVICE-BYVALUE instance of the DIRSRV macro instruction is used to obtain a protocol and service name pair when a service number is known. The information returned is obtained locally from the service configuration member (DNRSRVxx).

```
[symbol] DIRSRV GET,SERVICE,BYVALUE,  
          NABUF = name_address,  
          NALEN = name_length,  
          VABUF = value_address,  
          VALEN = value_length  
          [,SYSID = MVS_subsystem_id]  
          [,SIZE = size_limit]  
          [,ECB = INTERNAL | event_control_block_addr]  
          [,EXIT = exit_routine_address]  
          [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,SERVICE,BYVALUE</b>	Indicates that DNR is to search for and return a list of protocol and service name pairs associated with the service port number provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the DNR returns a list of protocol and service name pairs. The protocol and service name entries are separated by a slash (/). Each entry in the list is separated by a space character. The number of entries returned is returned in the SIZE operand.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the storage area identified by the NABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no value returned)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which the application program has placed a service port number.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the port number located in the storage area identified by the VABUF operand. This number should be 2 (bytes). A length of zero is invalid.  Default: 0 (no value returned)
<b>SYSID = <i>MVS_subsystem_id</i></b>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)

<b>SIZE</b> = <i>size_limit</i>	<p>Indicates a limit of the number of protocol and service name pairs returned. A value of 0 in the request, indicates there is no limit and the DNR is to return all name pairs associated with the port number. If the return information would not fit in the storage area provided or if the application program specifies a limit less than the defined number of protocol and service name pairs, the DCMORE conditional completion code is returned.</p> <p>Default: 0 (no size limit)</p>
<b>ECB</b> = INTERNAL   <i>event_control_block_addr</i>	<p>Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.</p> <p>The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:</p> <p>If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.</p> <p>If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.</p> <p>If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.</p> <p>Default: INTERNAL (internal ECB)</p>
<b>EXIT</b> = <i>exit_routine_address</i>	<p>Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.</p> <p>Default: not indicated (no exit routine)</p>
<b>MF</b> = (I   L   G   M   E, [ <i>dpl_address</i> ])	<p>Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.</p> <p><b>Note:</b> It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.</p> <p>Default: MF=I (inline, nonreentrant)</p>

## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. If the application program specified a SIZE limit less than the defined number of entries for the given host, or if the entire list of return information would not fit in the storage area provided, the DCMORE conditional completion code is returned.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

## Return Codes

This table lists the symbolic return codes for the GET-SERVICE-BYVALUE macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-10 GET-SERVICE-BYVALUE Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY		
DRFAILED	DAEXCPTN	DENONAME	DENOVALU	DENOTFND
		DENOCDS	DEOVRFLO	DENAMODE
		DEVAMODE		
DRFATLPTL	DAENVIRO	DESYSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
		DETERM		
DRFATLPTL	DAFORMAT	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
		DADPLERR	DEBDTYPE	DEPROTCT

## Example

This example demonstrates the use of GET-SERVICE-BYVALUE. The request is to find the protocol and service name pair for the service port number 9.

The application supplies this DIRSRV information:

NABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**GET-SERVICE-BYVALUE**

---

NALEN = 100  
VABUF = (address of:)

9																	
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1  
This information is returned:  
NABUF = (address of:)

T	C	P	/	D	I	S	C	A	R	D		U	D	P	/		
D	I	S	C	A	R	D											

NALEN = 23  
VABUF = (address of:)

9																	
---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 1



---

## GET-HOSTINFO-BYNAME

The GET-HOSTINFO-BYNAME instance of the DIRSRV macro instruction is used to obtain CPU and operating system information for a given host when its name is known. The name provided may be a local alias or a partial or fully-qualified domain name. The information returned is obtained globally from the Internet Domain Name System (DNS).

```
[symbol] DIRSRV GET,HOSTINFO,BYNAME,  
      NABUF = name_address,  
      NALEN = name_length,  
      VABUF = value_address,  
      VALEN = value_length  
      [,QNBUF = qualified_name_address]  
      [,QNLLEN = qualified_name_length]  
      [,SYSID = MVS_subsystem_id]  
      [,TIME = time_limit]  
      [,OPTCD = ([SYNC | ASync]  
                [,BLOCK | NOBLOCK]  
                [,COPY | ORIGINAL]))]  
      [,ECB = INTERNAL | event_control_block_addr]  
      [,EXIT = exit_routine_address]  
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,HOSTINFO,BYNAME</b>	Indicates that DNR is to search for and return CPU and operating system information associated with the host name or alias provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the name of a host. The name may be a local alias defined in the alias configuration member (DNRALCxx) or an Internet domain name or alias name defined by the DNS. Internet domain names may be partially or fully qualified. All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a fully-qualified domain name is 255 bytes. The maximum length of a local alias name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which DNR returns the CPU and operating system information. The strings are EBCIDIC strings separated by a space character.  Default: 0 (no value storage area)

<b>VALEN</b> = <i>value_length</i>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no value returned)
<b>QNBUF</b> = <i>qualified_name_address</i>	Indicates the address of a storage area in which DNR returns the fully qualified name used to search for the requested information. The fully qualified name is either the result of a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.  Default: 0 (no qualified name storage area)
<b>QNLEN</b> = <i>qualified_name_length</i>	Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully-qualified name returned. If QNLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.  Default: 0 (no qualified name returned)
<b>SYSID</b> = <i>MVS_subsystem_id</i>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)
<b>TIME</b> = <i>time_limit</i>	Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, DETIMOUT error code is returned the application program.  Default: 0 (use MAXTIME limit in DNRCFG00)

---

**OPTCD = SYNC | ASYNC**

Indicates the synchronization mode to be used when executing this macro instruction.

If the option is **OPTCD=SYNC**, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.

If the option is **OPTCD=ASYNC**, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The **ECB** and **EXIT** operands determine the form of asynchronous notification.

Default: **SYNC** (synchronous mode)

**OPTCD = BLOCK | NOBLOCK**

Indicates the disposition of a request if the information is not immediately available.

If the option is **OPTCD=BLOCK**, processing continues in accordance with the synchronization mode selected (**OPTCD=SYNC | ASYNC**). This option affects disposition of a request after it has been accepted by the DNR.

If the option is **OPTCD=NOBLOCK**, the request is completed abnormally and the specific error code is set to **DENOBLOK** indicating that the request could not be completed immediately. **OPTCD=NOBLOCK** is generally used in conjunction with **OPTCD=SYNC** to prevent suspension of the issuing task for an extended period of time.

Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (**OPTCD=ASYNC**).

Default: **BLOCK** (suspend task indefinitely)

**OPTCD = COPY | ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is **OPTCD=COPY**, a local copy of the requested information may be used to satisfy the request. Specifying **OPTCD=COPY** lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations.

If the option is **OPTCD=ORIGINAL**, the information must be obtained from its original, authoritative source. Specifying **OPTCD=ORIGINAL** assures that the most current information is returned.

Default: **COPY** (use copy of original data)

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in EXIT = exit\_routine\_address.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
*[dpl\_address]*)

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALen operands is not the fully qualified name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the fully qualified name in the QNBUF storage area. If the fully qualified name could not fit in the storage area, DCOVERFLO is set.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

If the fully qualified search string did not conform to the syntactic rules described in Syntactic Rules for Names, a DEBDNAME error code is returned. If the fully qualified search string is a valid host name but the host does not exist, the DNR returns an error code of DENAMERR. If the fully qualified search string is a valid and existing host but there is no specific data configured to satisfy the request, the DNR returns an error code of DENODATA. The fully qualified search string is the result of a fully qualified name given in the NABUF storage area, a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.

## Return Codes

This table lists the symbolic return codes for the GET-HOSTINFO-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-11 GET-HOSTINFO-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY	DCALIAS	DCOVERFLO
DRFAILED	DAEXCPTN	DENONAMEDETI	DENOVALU	DENOQNAME
		MOUT DENOCDS	DERFAIL	TFND DECOVERFLO
		DENOBLOK	DENAMERR	DENAMODE
		DEVAMODE	DENODATA	DEQNMODE
	DAENVIRO	DESYSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
		DETERM		
	DAFORMAT	DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

**Usage Information**

The GET-HOSTINFO-BYNAME instance of the DIRSRV macro is used by application programs to obtain CPU and operating system information. The DNR returns information received from DNS Host Information (HINFO) records. HINFO records specify the data returned by the DNR.

**Example**

This example demonstrates the use of GET-HOSTINFO-BYNAME. The request is to obtain CPU and operating system information about UNIX.

The alias configuration member (DNRALCxx) contains:

ACSS ZEUS.ACC.COM.  
UNIX SALT.

The application supplies this DIRSRV information:

NABUF = (address of:)

U	N	I	X																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 4

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

QNBUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

QNLEN = 100

This information is returned:

NABUF = (address of:)

U	N	I	X																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 4

VABUF = (address of:)

V	A	X	-	1	1	/	7	8	5		U	N	I	X					
---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	--	--	--	--	--

VALEN = 15

QNBUF = (address of)

S	A	L	T	.	A	C	C	.	C	O	M	.							
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--

QNLEN = 13

SALT.ACC.COM. was used to search the directory because UNIX was found as an alias in the alias configuration member. Because the replacement string, SALT, was partially qualified, the search list strings were appended to SALT.

---

ACC.COM. was included in the search list because the subsystem name ACSS is included in the alias configuration member. The DNR received a DNS response for SALT.ACC.COM. and returned the information to the application program.

## GET-HOSTSERV-BYNAME

The GET-HOSTSERV-BYNAME instance of the DIRSRV macro instruction is used to obtain a list of well-known service supported by a host. The name provided may be a local alias or a partial or fully-qualified domain name. The information returned is obtained globally from the Internet Domain Name System (DNS).

```
[symbol] DIRSRV GET,HOSTSERV,BYNAME,
      NABUF = name_address,
      NALEN = name_length,
      VABUF = value_address,
      VALEN = value_length
      [,QNBUF=qualified_name_address]
      [,QNLN=qualified_name_length]
      [,SYSID=MVS_subsystem_id]
      [,TIME=time_limit]
      [,SIZE=size_limit]
      [,OPTCD = ([SYNC | ASYNC]
                [,BLOCK | NOBLOCK]
                [,COPY | ORIGINAL])]
      [,ECB = INTERNAL | event_control_block_addr]
      [,EXIT = exit_routine_address]
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,HOSTSERV,BYNAME</b>	Indicates that DNR is to search for and return a list of well-known services supported by the host name or alias provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the name of a host. The name may be a local alias defined in the alias configuration member (DNRALCxx) or an Internet domain name or alias name defined by the DNS. Internet domain names may be partially or fully qualified. All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a fully-qualified domain name is 255 bytes. The maximum length of a local alias name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)



---

<b>VABUF</b> = <i>value_address</i>	<p>Indicates the address of a storage area in which DNR returns a list of well-known services supported by the host. The list contains a four-byte Internet address followed by a two-byte protocol number, a two-byte integer indicating the number of ports supported, followed by an array of two-byte ports. The array size is indicated by the two-byte number preceding the array. Each entry describes the well-known services supported by a particular protocol on a particular internet address. The number of entries is the smaller of the total number defined for the host, the length of the storage area available, or the number indicated by the SIZE operand.</p> <p>Default: 0 (no value storage area)</p>
<b>VALEN</b> = <i>value_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.</p> <p>Default: 0 (no value returned)</p>
<b>QNBUF</b> = <i>qualified_name_address</i>	<p>Indicates the address of a storage area in which DNR returns the fully qualified name used to search for the requested information. The fully qualified name is either the result of a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.</p> <p>Default: 0 (no qualified name storage area)</p>
<b>QNLEN</b> = <i>qualified_name_length</i>	<p>Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully-qualified name returned. If QNLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.</p> <p>Default: 0 (no qualified name returned)</p>
<b>SYSID</b> = <i>MVS_subsystem_id</i>	<p>Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.</p> <p>Default: not indicated (use installation default)</p>

**TIME** = *time\_limit*

Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, DETIMOUT error code is returned the application program.

Default: 0 (use MAXTIME limit in DNRCFG00)

**SIZE** = *size\_limit*

Indicates a limit of the number of entries returned. A value of zero in the request, indicates there is no limit and the DNR is to return all entries associated with the domain name. If the return information would not fit in the storage area provided or if the application program specifies a limit less than the defined number of entries for the given host, the DCMORE conditional completion code is returned.

Default: 0 (no size limit)

---

**OPTCD = SYNC | ASYNC**

Indicates the synchronization mode to be used when executing this macro instruction.

If the option is **OPTCD=SYNC**, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.

If the option is **OPTCD=ASYNC**, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The **ECB** and **EXIT** operands determine the form of asynchronous notification.

Default: **SYNC** (synchronous mode)

**OPTCD = BLOCK | NOBLOCK**

Indicates the disposition of a request if the information is not immediately available.

If the option is **OPTCD=BLOCK**, processing continues in accordance with the synchronization mode selected (**OPTCD=SYNC | ASYNC**). This option code affects disposition of a request after it has been accepted by the DNR.

If the option is **OPTCD=NOBLOCK**, the request is completed abnormally and the specific error code is set to **DENOBLOK** indicating that the request could not be completed immediately. **OPTCD=NOBLOCK** is generally used in conjunction with **OPTCD=SYNC** to prevent suspension of the issuing task for an extended period of time.

Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (**OPTCD=ASYNC**).

Default: **BLOCK** (suspend task indefinitely)

**OPTCD = COPY | ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is **OPTCD=COPY**, a local copy of the requested information may be used to satisfy the request. Specifying **OPTCD=COPY** lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations.

If the option is **OPTCD=ORIGINAL**, the information must be obtained from its original, authoritative source. Specifying **OPTCD=ORIGINAL** assures that the most current information is returned.

Default: **COPY** (use copy of original data)

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
*dpl\_address*)

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALEN operands is not the fully qualified name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the fully qualified name in the QNBUF storage area. If the fully qualified name could not fit in the storage area, DCOVERFLO is set. If the application program specified a SIZE limit less than the defined number of entries for the given host, or if the entire list of return information would not fit in the storage area provided, the DCMORE conditional completion code is returned.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

If the fully qualified search string did not conform to the syntactic rules described in Syntactic Rules for Names, a DEBDNAME error code is returned. If the fully qualified search string is a valid host name but the host does not exist, the DNR returns an error code of DENAMERR. If the fully qualified search string is a valid and existing host but there is no specific data configured to satisfy the request, the DNR returns an error code of DENODATA. The fully qualified search string is the result of a fully qualified name given in the NABUF storage area, a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.

## Return Codes

This table lists the symbolic return codes for the GET-HOSTSERV-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-12 GET-HOSTSERV-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY DCOVERFLO	DCMORE	DCALIAS
DRFAILED	DAEXCPTN	DENONAME	DENOVALU	DENOQNAM
		DETIMOUT	DERFAIL	DENOTFND
		DENOCDS	DENAMERR	DEOVRFLO
		DENOBLOK	DENODATA	DENAMODE
	DAENVIRO	DEVAMODE	DEQNMODE	
		DESYSERR	DESUBSYS	DENOTCNF
		DENOTACT	DENOTRDY	DESTOP
		DEUNAVBL	DERSOURC	DENOTPRB
	DAFORMAT	DETERM		
		DEBDOPCD	DEBDFNCD	DEBDXECB
		DEBDEXIT	DEBDNAME	DEBDVALU

**Table 2-12 GET-HOSTSERV-BYNAME Return Codes (Continued)**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

**Usage Information**

The GET-HOSTSERV-BYNAME instance of the DIRSRV macro is used by application programs to determine the services supported by a given Internet domain name. The DNR returns information received from DNS Well Known Services (WKS) records. WKS records specify the well-known services supported by a particular protocol on a particular internet address. The two-byte protocol number returned in an entry may be used as input in a DIRSRV GET-PROTOCOL-BYVALUE request to obtain the associated protocol name. Likewise, the two byte port numbers may be used as input in a DIRSRV GET-SERVICE-BYVALUE request to obtain the associated service names.

**Example**

This example demonstrates the use of GET-HOSTSERV-BYNAME. The request is to find the list of well-known services supported by host TERP.UMD.EDU.

The application supplies this DIRSRV information:

NABUF = (address of:)

T	E	R	P	.	U	M	D	.	E	D	U				
---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

NALEN = 12

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

QNBUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

QNLEN = 100

SIZE = 100

This information is returned:

NABUF = (address of:)

T	E	R	P	.	U	M	D	.	E	D	U	.			
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

NALEN = 13

VABUF = (address of:)

0x80	0x08	0x0a	0x5a		17		2		53		123				
0x80	0x08	0x0a	0x5a		6		3		21		23				
		25													



## GET-ROUTE-BYNAME

The GET-ROUTE-BYNAME instance of the DIRSRV macro instruction is used to obtain a list of hosts willing to act as a mail exchange for a host. The name provided may be a local alias or a partial or fully qualified domain name. The information returned is obtained globally from the Internet Domain Name System (DNS).

```
[symbol] DIRSRV GET,ROUTE,BYNAME,
      NABUF = name_address,
      NALEN = name_length,
      VABUF = value_address,
      VALEN = value_length
      [,QNBUF = qualified_name_address]
      [,QNLEN = qualified_name_length]
      [,SYSID = MVS_subsystem_id]
      [,TIME = time_limit]
      [,SIZE = size_limit]
      [,OPTCD = ([SYNC | ASync]
                [,BLOCK | NOBLOCK]
                [,COPY | ORIGINAL])]
      [,ECB = INTERNAL | event_control_block_addr]
      [,EXIT = exit_routine_address]
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,ROUTE,BYNAME</b>	Indicates that DNR is to search for and return a list of host names willing to act as mail exchanges for the host name or alias provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed the name of a host. The name may be a local alias defined in the alias configuration member (DNRALCxx) or an Internet domain name or alias name defined by the DNS. Internet domain names may be partially or fully qualified. All names must conform to the syntactic rules described in Syntactic Rules for Names.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of a fully-qualified domain name is 255 bytes. The maximum length of a local alias name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which DNR returns a list of hosts willing to act as mail exchanges. The list contains EBCDIC character strings separated by a space character. The number of strings returned is specified in the SIZE operand. The list is sorted by preference giving the highest preferred host first. The number of entries is the smaller of the total number defined for the host, the length of the storage area available, or the number indicated by the SIZE operand.  Default: 0 (no value storage area)



---

<b>VALEN</b> = <i>value_length</i>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no value returned)
<b>QNBUF</b> = <i>qualified_name_address</i>	Indicates the address of a storage area in which DNR returns the fully qualified name used to search for the requested information. The fully qualified name is either the result of a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.  Default: 0 (no qualified name storage area)
<b>QNLEN</b> = <i>qualified_name_length</i>	Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully-qualified name returned. If QNLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.  Default: 0 (no qualified name returned)
<b>SYSID</b> = <i>MVS_subsystem_id</i>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)
<b>TIME</b> = <i>time_limit_</i>	Indicates the maximum amount of elapsed time (in seconds) to find the requested information. Since portions of the distributed database are maintained on other systems, queries must sometimes be transmitted to remote destinations. Also, the responses to such queries may contain referrals to other remote systems. Therefore, some types of directory requests may take an arbitrary amount of time to complete. This operand can be used to limit the amount of time spent searching for specific information. If the directory search is abandoned because the time limit has been exceeded, DETIMOUT error code is returned the application program.  Default: 0 (use MAXTIME limit in DNRCFG00)
<b>SIZE</b> = <i>size_limit</i>	Indicates a limit of the number of entries returned. A value of zero in the request, indicates there is no limit and the DNR is to return all entries associated with the domain name. If the return information would not fit in the storage area provided or if the application program specifies a limit less than the defined number of entries for the given host, the DCMORE conditional completion code is returned.  Default: 0 (no size limit)

**OPTCD = SYNC | ASYNC** Indicates the synchronization mode to be used when executing this macro instruction.

If the option is **OPTCD=SYNC**, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.

If the option is **OPTCD=ASYNC**, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.

Default: SYNC (synchronous mode)

**OPTCD = BLOCK | NOBLOCK**

Indicates the disposition of a request if the information is not immediately available.

If the option is **OPTCD=BLOCK**, processing continues in accordance with the synchronization mode selected (**OPTCD=SYNC | ASYNC**). This option code affects disposition of a request after it has been accepted by the DNR.

If the option is **OPTCD=NOBLOCK**, the request is completed abnormally and the specific error code is set to **DENOBLOK** indicating that the request could not be completed immediately. **OPTCD=NOBLOCK** is generally used in conjunction with **OPTCD=SYNC** to prevent suspension of the issuing task for an extended period of time.

Note The issuing task may be momentarily suspended to let the DNR address space process the request. If the application program cannot afford to be suspended, even for a very small amount of time, then the request must be executed asynchronously (**OPTCD=ASYNC**).

Default: BLOCK (suspend task indefinitely)

**OPTCD = COPY | ORIGINAL**

Indicates whether a copy of the information can be used to complete the request or whether original information must be returned.

If the option is **OPTCD=COPY**, a local copy of the requested information may be used to satisfy the request. Specifying **OPTCD=COPY** lets information obtained as the result of a previous request be reused and returned to the application program, thus avoiding time-consuming queries to remote destinations.

If the option is **OPTCD=ORIGINAL**, the information must be obtained from its original, authoritative source. Specifying **OPTCD=ORIGINAL** assures that the most current information is returned.

Default: COPY (use copy of original data)

---

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in EXIT = exit\_routine\_address.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
*[dpl\_address]*)

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

## Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALLEN operands is not the fully qualified name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the fully qualified name in the QNBUF storage area. If the fully qualified name could not fit in the storage area, DCOVRFLO is set. If the application program specified a SIZE limit less than the defined number of entries for the given host, or if the entire list of return information would not fit in the storage area provided, the DCMORE conditional completion code is returned.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

If the fully qualified search string did not conform to the syntactic rules described in Syntactic Rules for Names, a DEBDNAME error code is returned. If the fully qualified search string is a valid host name but the host does not exist, the DNR returns an error code of DENAMERR. If the fully qualified search string is a valid and existing host but there is no specific data configured to satisfy the request, the DNR returns an error code of DENODATA. The fully qualified search string is the result of a fully qualified name given in the NABUF storage area, a local alias lookup, a name formed by appending the DNR search list strings to a partially qualified domain name, or a DNS alias referral.

### Return Codes

This table lists the symbolic return codes for the GET-ROUTE-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-13 GET-ROUTE-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY DCOVRFLO	DCMORE DCLOCAL	DCALIAS
DRFAILED	DAEXCPTN	DENONAMEDETI MOUT DENOCDS DENOBLOK DEVAMODE	DENOVALU DERFAIL DENAMERR DENODATA DEQNMODE	DENOQNAM DENOTFND DEOVRFLO DENAMODE
	DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DESUBSYS DENOTRDY DERSOURC	DENOTCNF DESTOP DENOTPRB
	DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME	DEBDXECB DEBDVALU
DRFATLPTL	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

---

## Usage Information

The GET-ROUTE-BYNAME instance of the DIRSRV macro is used by mailers to determine how to route a message given an Internet domain name. The DNR returns information received from DNS Mail Exchange (MX) records. MX records specify mail exchange hosts that know how to route mail to a host not directly connected to the Internet. MX records include a preference value that specifies the preference given to the host relative to other hosts. The DNR sorts MX records according to preference (lower values are preferred) and returns the host names in order. This order reflects the order the calling mailer programs should follow when delivering mail. To determine the internet address of the remote hosts, mailers should pass the hosts names returned in the GET-ROUTE-BYNAME request as search strings to a DIRSRV GET-HOST-BYNAME request.

## Example

This example demonstrates the use of GET-ROUTE-BYNAME. The request is to find the list of hosts willing to act as a mail exchange for the host UNIX. The alias configuration member (DNRALCxx) contains:

UNIX VAX.ACC.COM.

The application supplies this DIRSRV information:

NABUF = (address of:)

U	N	I	X																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 4

VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100

QNBUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

QNLEN = 100

SIZE = 100

This information is returned:

NABUF = (address of:)

U	N	I	X																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 4

VABUF = (address of:)

S	A	T	U	R	N	.	A	C	C	.	C	O	M	.					
S	A	L	T	.	A	C	C	.	C	O	M	.							

## GET-ROUTE-BYNAME

---

VALEN = 29

QNBUF = (address of:)

S	A	T	U	R	N	.	A	C	C	.	C	O	M	.	
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

QNLEN = 15

SIZE = 2

SATURN.ACC.COM. is returned in the QNBUF because the search string UNIX was listed as an alias in the alias configuration member. The replacement string VAX.ACC.COM. was in turn known as an alias of SATURN.ACC.COM. in the domain name space. The DNR received a DNS response for SATURN.ACC.COM. and returned the information to the application program.

---

## GET-RPC-BYNAME

The GET-RPC-BYNAME instance of the DIRSRV macro instruction is used to obtain an RPC number when its RPC name is known. This information returned is obtained locally from the RPC configuration member (DNRRPCxx).

```
[symbol] DIRSRV GET,RPC,BYNAME,  
      NABUF = name_address,  
      NALEN = name_length,  
      VABUF = value_address,  
      VALEN = value_length  
      [,QNBUF = qualified_name_address]  
      [,QNLN = qualified_name_length]  
      [,SYSID = MVS_subsystem_id]  
      [,OPTCD = SYNC | ASYNC]  
      [,ECB = INTERNAL | event_control_block_addr]  
      [,EXIT = exit_routine_address]  
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,RPC,BYNAME</b>	Indicates that the DNR is to search for and return an RPC value associated with the RPC name provided by the application program.
<b>NABUF = <i>name_address</i></b>	Indicates the address of a storage area in which the application program has placed an RPC name. The name may be an RPC alias defined in the RPC configuration member (DNRRPCxx) or an official RPC name.  Default: 0 (no name storage area)
<b>NALEN = <i>name_length</i></b>	Indicates the length (in bytes) of the name located in the storage area identified by the NABUF operand. The maximum length of an RPC name is 40 bytes. A length of zero is invalid.  Default: 0 (no name provided)
<b>VABUF = <i>value_address</i></b>	Indicates the address of a storage area in which DNR returns an RPC value associated with the RPC name. Each returned RPC value is four bytes.  Default: 0 (no value storage area)
<b>VALEN = <i>value_length</i></b>	Indicates the length (in bytes) of the storage area identified by the VABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged. The minimum length of the storage area is 4 bytes.  Default: 0 (no value returned)
<b>QNBUF = <i>qualified_name_address</i></b>	Indicates the address of a storage area in which DNR returns the official RPC name used to search for the requested information. The DNR returns an RPC name in the QNBUF data area if the search string was an alias.  Default: 0 (no qualified name storage area)

<b>QNLLEN</b> = <i>qualified_name_length</i>	Indicates the length (in bytes) of the storage area identified by the QNBUF operand. This length is the maximum length of the storage area and is updated when the request is completed to reflect the actual length of the fully qualified name returned. If QNLLEN is non-zero, QNBUF must also be non-zero and point to the beginning of a valid storage area.  Default: 0 (no qualified name returned)
<b>SYSID</b> = <i>MVS_subsystem_id</i>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)
<b>OPTCD</b> = SYNC   ASYNC	Indicates the synchronization mode to be used when executing this macro instruction.  If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.  If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.  Default: SYNC (synchronous mode)



---

**ECB** = INTERNAL |  
*event\_control\_block\_addr*

Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.

The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:

If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in EXIT = exit\_routine\_address.

If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.

If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.

Default: INTERNAL (internal ECB)

**EXIT** = *exit\_routine\_address*

Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.

Default: not indicated (no exit routine)

**MF** = (I | L | G | M | E,  
*dpl\_address*)

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, dpl address, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

**Completion Information**

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly. DCALIAS is set if the name referenced in the NABUF and NALEN operands is not the official RPC name associated with the returned data. If DCALIAS is set and a storage area was supplied in the QNBUF and QNLEN operands, the DNR returns the official RPC name in the QNBUF storage area. If the official RPC name could not fit in the storage area, DCOVERFLO is set.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

**Return Codes**

This table lists the symbolic return codes for the GET-RPC-BYNAME macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-14 GET-RPC-BYNAME Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY	DCALIS	DCOVERFLO
DRFAILED	DAEXCPTN	DENONAMEDENOT FND DENAMODE	DENOVALUD ENOCDS DEVAMODE	DENOQNAMDE OVRFLO DEQNMODE
		DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DENOTCNF DESTOP DENOTPRB
DRFATLPTL	DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME	DEBDXECB DEBDVALU
	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

**Example**

This example demonstrates the use of GET-RPC-BYNAME. The request is to find the RPC number for the RPC name portmapper.

The application supplies this DIRSRV information:

NABUF = (address of:)

S	U	N	R	P	C														
---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--

---

NALEN = 6  
VABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

VALEN = 100  
QNBUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

QNLEN = 100  
SIZE = 0

This information is returned:

NABUF = (address of:)

S	U	N	R	P	C												
---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 6  
VABUF = (address of:)

0x00	0x01	0x86	0xa0					
------	------	------	------	--	--	--	--	--

VALEN = 4  
QNBUF = (address of:)

P	O	R	T	M	A	P	P	E	R								
---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--	--	--

QNLEN = 10

Because the search string, SUNRPC is given as an alias in the RPC configuration file, the official RPC name is returned in the qualified name buffer.

## GET-RPC-BYVALUE

The GET-RPC-BYVALUE instance of the DIRSRV macro instruction is used to obtain an RPC name when its number is known. The information returned is obtained locally from the RPC configuration member (DNRRPCxx).

```
[symbol] DIRSRV GET,RPC,BYVALUE,
      NABUF = name_address,
      NALEN = name_length,
      VABUF = value_address,
      VALEN = value_length
      [,SYSID = MVS_subsystem_id]
      [,OPTCD = SYNC | ASYNC]
      [,ECB = INTERNAL | event_control_block_addr]
      [,EXIT = exit_routine_address]
      [,MF = (I | L | G | M | E, [dpl_address])]
```

### Syntax Description

<b>GET,RPC,BYVALUE</b>	Indicates that the DNR is to search for and return an RPC name associated with the RPC number provided by the application program.
<b>NABUF = name_address</b>	Indicates the address of a storage area in which the DNR returns an RPC name.  Default: 0 (no name storage area)
<b>NALEN = name_length</b>	Indicates the length (in bytes) of the storage area identified by the NABUF operand. The length is updated when the request is completed to reflect the actual amount of information returned. If the request completes abnormally, no information is returned and the length remains unchanged.  Default: 0 (no name returned)
<b>VABUF = value_address</b>	Indicates the address of a storage area in which the application program has placed a four-byte RPC number.  Default: 0 (no value storage area)
<b>VALEN = value_length</b>	Indicates the length (in bytes) of the RPC number located in the storage area identified by the VABUF operand. The value of the storage area must be 4 (bytes).  Default: 0 (no value provided)
<b>SYSID = MVS_subsystem_id</b>	Indicates the ID of the MVS subsystem that is to process this request. Normally this operand is not required and an installation default is used. However, if more than one DNR subsystem is active on the local system, the particular subsystem that is to process this request must be indicated. The MVS subsystem ID is coded as an alphanumeric string up to four (4) characters in length.  Default: not indicated (use installation default)

---

<b>OPTCD = SYNC   ASYNC</b>	<p>Indicates the synchronization mode to be used when executing this macro instruction.</p> <p>If the option is OPTCD=SYNC, the request is executed in synchronous mode, and control is not returned to the application program until the requested macro instruction is complete.</p> <p>If the option is OPTCD=ASYNC, the request is executed in asynchronous mode, and control is returned to the application program immediately after it has been accepted by DNR. The application program is notified asynchronously when the request has been completed. The ECB and EXIT operands determine the form of asynchronous notification.</p> <p>Default: SYNC (synchronous mode)</p>
<b>ECB = INTERNAL   <i>event_control_block_addr</i></b>	<p>Indicates the location of an event control block (ECB) to be posted by DNR when the directory request has been completed. The ECB can be any fullword of storage aligned on a fullword boundary.</p> <p>The ECB and EXIT operands share the same storage location in the Directory Services Parameter List (DPL), and are therefore mutually exclusive operands. If asynchronous mode has been specified (OPTCD=ASYNC), the ECB-EXIT field of the DPL (DPLECBXR) is used in this manner:</p> <p>If the option is ECB=address, DNR uses the field as the address of an external ECB. The application program is responsible for issuing a WAIT macro instruction and clearing the ECB when posted. If EXIT=address is specified, DNR uses the field as the address of an exit routine, and schedules the routine as indicated in the following EXIT operand text.</p> <p>If the option is ECB=INTERNAL, DNR uses the field as an internal ECB. The application program is responsible for issuing a WAIT macro instruction specifying the internal ECB, but need not clear it when posted.</p> <p>If synchronous mode has been specified (OPTCD=SYNC), the parameter list is flagged to be processed as if ECB=INTERNAL had been specified, and the ECB-EXIT field is used as an internal ECB that is waited on and cleared automatically before control is returned to the application program.</p> <p>Default: INTERNAL (internal ECB)</p>
<b>EXIT = <i>exit_routine_address</i></b>	<p>Indicates the address of a routine to be scheduled when the request is completed. The EXIT and ECB operands share the same storage location in the parameter list and are therefore mutually exclusive. The completion exit is scheduled only if asynchronous mode has been indicated by OPTCD=ASYNC. If synchronous mode has been indicated, the exit routine is not used. If one is specified with this operand, the address is overwritten with an internal ECB before the request completes.</p> <p>Default: not indicated (no exit routine)</p>

**MF** = (I | L | G | M | E,  
[*dpl\_address*])

Indicates the standard (inline), list, generate, modify, or execute form of the DIRSRV macro instruction. The second sublist operand, *dpl address*, specifies the address of the parameter list to be used for this request. If no MF operand is specified, the standard form is used.

**Note:** It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.

Default: MF=I (inline, nonreentrant)

### Completion Information

The DIRSRV macro instruction completes normally (or conditionally) when the requested information has been returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to 0 (DROKAY), and the conditional completion code is returned in register 0. The DPL return code field is set accordingly.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure.

- If the general return code is set to DRFAILED, the recovery action code is returned in register 0 and the DPL return code contains a specific error code that identifies a particular error.
- If the general return code is set to DRFATLPTL, the recovery action code and the error code are both returned in register 0 and the DPL is not updated.

### Return Codes

This table lists the symbolic return codes for the GET-RPC-BYVALUE macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-15 GET-RPC-BYVALUE Return Codes**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code		
DROKAY	DAOKAY	DCOKAY		
DRFAILED	DAEXCPTN	DENONAMEDENO CDS DEVAMODE	DENOVALU DEOVRFLO	DENOTFND DENAMODE
	DAENVIRO	DESYSERR DENOTACT DEUNAVBL DETERM	DESUBSYS DENOTRDY DERSOURC	DENOTCNF DESTOP DENOTPRB
DRFATLPTL	DAFORMAT	DEBDOPCD DEBDEXIT	DEBDFNCD DEBDNAME	DEBDXECB DEBDVALU
	DAPROCED	DEACTIVE		
	DADPLERR	DEBDTYPE	DEPROTCT	DEPLMODE

---

### Example

This example demonstrates the use of GET-RPC-BYVALUE. The request is to find the RPC name for the RPC value 100000.

The application supplies this DIRSRV information:

NABUF = (address of:)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

NALEN = 100

VABUF = (address of:)

0x00	0x01	0x86	0xa0						
------	------	------	------	--	--	--	--	--	--

VALEN = 4

This information is returned:

NABUF = (address of:)

P	O	R	T	M	A	P	P	E	R						
---	---	---	---	---	---	---	---	---	---	--	--	--	--	--	--

NALEN = 10

VABUF = (address of:)

0x00	0x01	0x86	0xa0						
------	------	------	------	--	--	--	--	--	--

VALEN = 4

## PURGE

The PURGE instance of the DIRSRV macro instruction is used to purge a previously issued asynchronous DIRSRV request. The DNR searches for the DPL and completes the request.

**[symbol] DIRSRV PURGE,**  
**MF = (E, *dpl\_address*)**

### Syntax Description

PURGE	Indicates that the DNR is to purge a previously issued DIRSRV request.
MF = (E, [ <i>dpl_address</i> ])	Indicates the execute form of the PURGE macro instruction. The second sublist operand, <i>dpl address</i> , specifies the address of the parameter list to purge.  <b>Note:</b> It is recommended to use the list form of the macro to ensure DPL alignment on a full word boundary. If you do not use the list form, make sure you align the storage area used for the remote parameter list on a fullword boundary. This will ensure that the internal ECB is aligned on a fullword boundary.  Default: MF=E

### Return Codes

This table lists the symbolic return codes for the PURGE macro. The values associated with the symbolic names can be found in the DPL macro expansion.

**Table 2-16 PURGE Return Codes:**

General Return Code	Recovery Action Code	Conditional Or Specific Error Code
DROKAY	DAOKAY	DCOKAY
DRFAILED	DAENVIRO	DESYSERR