IBM

System i

Networking
Domain Name System

Version 6 Release 1

IBM

System i

Networking
Domain Name System

Version 6 Release 1

> **Note**
>
> Before using this information and the product it supports, read the information in "Notices," on page 43.

Contents

Domain Name System

Domain Name System (DNS) is a distributed database system for managing host names and their associated Internet Protocol (IP) addresses.

With DNS, people can use simple names, such as www.jkltoys.com to locate a host, rather than using the IP addresses, for example, 192.168.12.88 in IPv4, or 2001:D88::1 in IPv6. A single server might be responsible only for knowing the host names and IP addresses for a small part of a zone, but DNS servers can work together to map all domain names to their IP addresses. DNS servers that work together allow computers to communicate across the Internet.

For IBM® i5/OS® Version 6 Release 1 (V6R1), DNS services are based on the industry-standard DNS implementation, known as Berkeley Internet Name Domain (BIND) version 9. For previous i5/OS releases, DNS services were based on BIND version 8.2.5. To use the new BIND version 9 DNS server, you must have i5/OS option 31 (DNS) and option 33 (Portable Application Solutions Environment (PASE)) installed on your IBM System i® model. As of i5/OS V6R1, for security reasons, BIND 4 and 8 are replaced with BIND 9. Thus, the migration to BIND 9 is required for your DNS server.

What's new for V6R1

Read about new or significantly changed information for the Domain Name System (DNS) topic collection.

BIND 9

The Berkeley Internet Name Domain (BIND) version 9, introduced this release, provides several features to enhance the performance of your Domain Name System (DNS) server. For example, it supports name-to-address and address-to-name lookups in all currently defined forms of IPv6. It uses the *view* statement, which allows a single DNS instance to answer a query differently depending on where the query comes from, such as the Internet or an intranet. In addition, it uses journal files to hold dynamic updates for a zone.

The previous BIND 4.9.3 and BIND 8.2.5 are no longer supported, and needed to be migrated to BIND 9.

New configuration commands

The following configuration commands were added to make it easier to manage those DNS configuration files on your system.

Create RNDC Configuration (CRTRNDCCFG)
> The RNDC Configuration Utility (CRTRNDCCFG) command is used to generate RNDC configuration files. It is a convenient alternative to writing the rndc.conf file and its corresponding controls and key statements in the named.conf file.

DNS Configuration Utility (CHKDNSCFG)
> The DNS Configuration Utility (CHKDNSCFG) command checks the syntax of the configuration file called named.conf. But it does not provide the support to check the semantics of the configuration file.

DNS Zone Utility (CHKDNSZNE)
> The DNS Zone Utility (CHKDNSZNE) command checks the syntax and integrity of a zone data file. It is helpful to check zone data files before you add them into a DNS server.

New query and update utilities

The following query and update utilities were added to enhance the management capabilities of your DNS server.

Domain Information Groper (DIG)
> You can use the DIG query tool to retrieve DNS information about hosts, domains, and other DNS servers based on the response of a DNS server. You can also use it to verify if a DNS server is responding correctly before you configure your system to use it.

Start HOST Query (HOST)
> The Start HOST Query (HOST) command is used for DNS lookups. It converts domain names to IP addresses (either IPv4 or IPv6) and vice versa.

Dynamic Update Utility (NSUPDATE)
> The Dynamic Update Utility (NSUPDATE) command submits Dynamic DNS Update requests as defined in Request for Comments (RFC) 2136 to a DNS server. This allows resource records to be added or removed from a zone while the DNS server is running. Thus, you do not need to update records by manually editing the zone file. A single update request can contain requests to add or remove more than one resource record, but the resource records that are dynamically added or removed with the NSUPDATE command should be in the same zone.

Remote Name Daemon Control (RNDC)
> The Remote Name Daemon Control (RNDC) command allows a system administrator to control the operation of a name server. It reads a configuration file, called *rndc.conf*, to determine how to contact the name server and to determine what algorithm and key it should use. If no rndc.conf file is found, then, by default, an rndc-key._KID file that is created during installation is used, which automatically grants access through the loopback interface.

How to see what's new or changed

To help you see where technical changes have been made, the information center uses:
* The » image to mark where new or changed information begins.
* The « image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

> **Related reference**
> "BIND 9 features" on page 8
> BIND 9 is similar to BIND 8; however, it provides several features to enhance performance of your Domain Name System (DNS) server, such as views.

PDF file for Domain Name System

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select Domain Name System (about 625 KB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:
1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe® Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

> **Related reference**
>
> "Related information for Domain Name System" on page 41
> IBM Redbooks® publications, Web sites, and other information center topic collections contain information that relates to the Domain Name System (DNS) topic collection. You can view or print any of the PDF files.

Domain Name System concepts

Domain Name System (DNS) is a distributed database system for managing host names and their associated Internet Protocol (IP) addresses. With DNS, you can use simple names, such as www.jkltoys.com, to locate a host, rather than using the IP addresses, for example, 192.168.12.88 in IPv4, or 2001:D88::1 in IPv6.

A single server might be responsible only for knowing the host names and IP addresses for a small part of a zone, but DNS servers can work together to map all domain names to their IP addresses. DNS servers that work together allows computers to communicate across the Internet.

DNS data is broken up into a hierarchy of domains. Servers are responsible to know only a small portion of data, such as a single subdomain. The portion of a domain for which the server is directly responsible is called a zone. A DNS server that has complete host information and data for a zone is authoritative for the zone. An authoritative server can answer queries about hosts in its zone, using its own resource records. The query process depends on a number of factors. Understanding DNS queries explains the paths that a client can use to resolve a query.

Understanding zones

Domain Name System (DNS) data is divided into manageable sets of data called *zones*. And each of these sets is a specific zone type.

Zones contain name and IP address information about one or more parts of a DNS domain. A server that contains all of the information for a zone is the authoritative server for the domain, called a *parent zone*. Sometimes it makes sense to delegate the authority for answering DNS queries for a particular subdomain to another DNS server, called a *child zone*. In this case, the DNS server for the domain can be configured to refer the subdomain queries to the appropriate server.

For backup and redundancy, zone data is often stored on servers other than the authoritative DNS server. These other servers are called secondary servers, which load zone data from the authoritative server. Configuring secondary servers allows you to balance the demand on servers and also provides a backup in case the primary server goes down. Secondary servers obtain zone data by doing zone transfers from the authoritative server. When a secondary server is initialized, it loads a complete copy of the zone data from the primary server. The secondary server also reloads zone data from the primary server or from other secondaries for that domain when zone data changes.

DNS zone types

You can use i5/OS DNS to define several types of zones to help you manage DNS data:

Primary zone
> A primary zone loads zone data directly from a file on a host. It can contain a subzone, or child zone. It can also contain resource records, such as host, alias (CNAME), IPv4 address (A), IPv6 address (AAAA), or reverse mapping pointer (PTR) records.

Note: Primary zones are sometimes referred to as *master zones* in other BIND documentation.

Subzone

A subzone defines a zone within the primary zone. Subzones allow you to organize zone data into manageable pieces.

Child zone

A child zone defines a subzone and delegates responsibility for the subzone data to one or more name servers.

Alias (CNAME)

An alias defines an alternate name for a primary domain name.

Host A host object maps A and PTR records to a host. Additional resource records can be associated with a host.

Secondary zone

A secondary zone loads zone data from a zone's primary server or another secondary server. It maintains a complete copy of the zone for which it is a secondary.

Note: Secondary zones are sometimes referred to as *slave zones* in other BIND documentation.

Stub zone

A stub zone is similar to a secondary zone, but it only transfers the name server (NS) records for that zone.

Forward zone

A forward zone directs all queries for that particular zone to other servers.

Related concepts

"Understanding Domain Name System queries"
Domain Name System (DNS) clients use DNS servers to resolve queries. The queries might come directly from the client or from an application running on the client.

Related tasks

"Configuring zones on a name server" on page 29
After you configure a Domain Name System (DNS) server instance, you need to configure the zones for the name server.

Related reference

"Example: Single Domain Name System server for an intranet" on page 14
This example depicts a simple subnet with a Domain Name System (DNS) server for internal use.

"Domain Name System resource records" on page 9
Resource records are used to store data about domain names and IP addresses. You can use the Resource record lookup table to look into the resource records supported for the i5/OS operating system.

Understanding Domain Name System queries

Domain Name System (DNS) clients use DNS servers to resolve queries. The queries might come directly from the client or from an application running on the client.

The client sends a query message to the DNS server that contains a fully qualified domain name (FQDN), a query type, such as a particular resource record the client requires, and the class for the domain name, which is typically the Internet (IN) class. The following figure depicts the sample network from the Single DNS server with Internet access example.
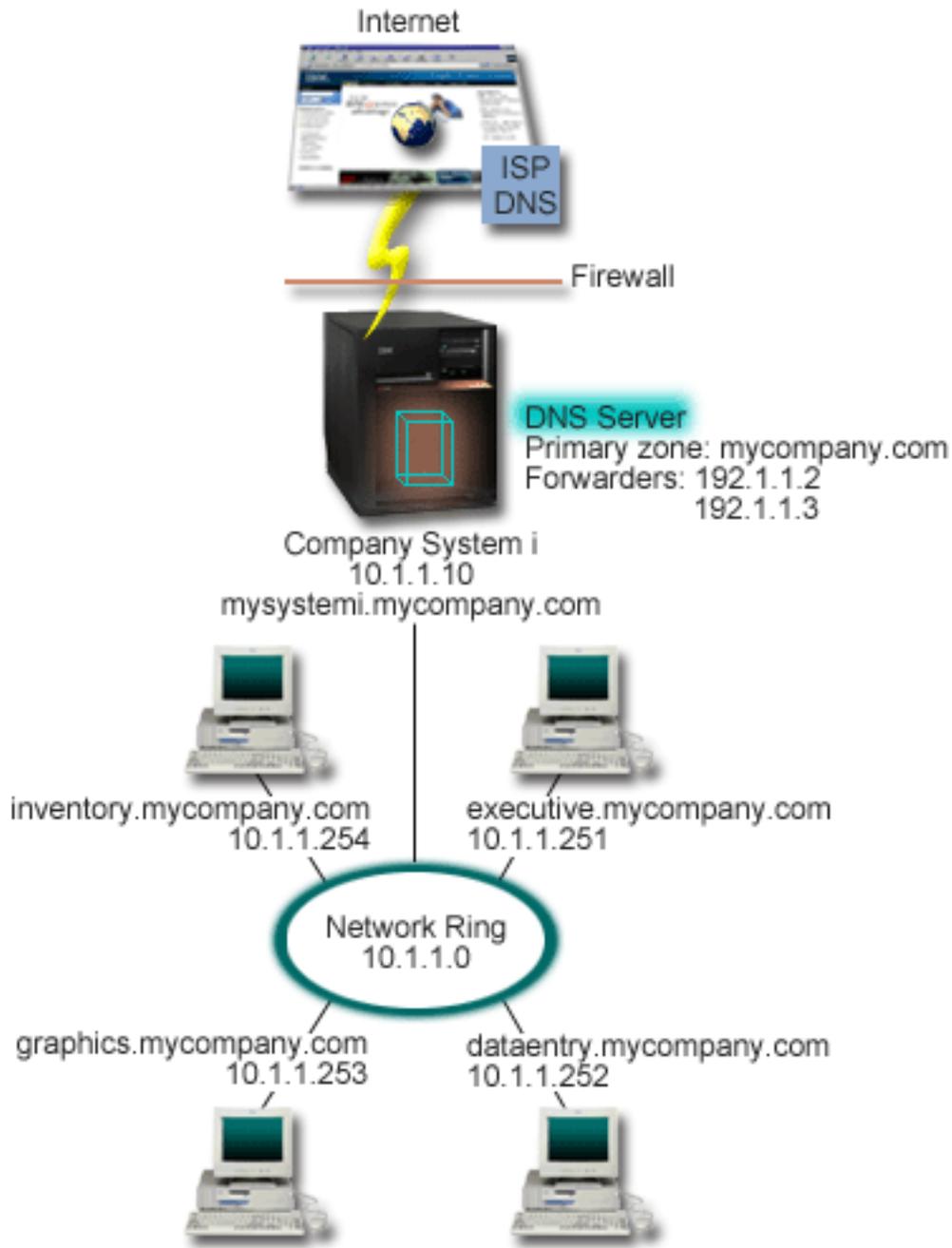
Figure 1. Single DNS server with Internet access

Suppose that host *dataentry* queries the DNS server for graphics.mycompany.com. The DNS server uses its own zone data and responds with the IP address 10.1.1.253.

Now suppose *dataentry* requests the IP address of www.jkl.com. This host is not in the DNS server's zone data. Two paths can be followed: *recursion* or *iteration*. If a DNS server is set to use *recursion*, the server can query or contact other DNS servers on behalf of the requesting client to fully resolve the name, and then send an answer back to the client. Additionally, the requesting server stores the answer into its cache so that the answer can be used the next time that the server receives that query. If a DNS server is set to use *iteration*, a client can attempt to contact other DNS servers on its own to resolve a name. In this process, the client uses separate and additional queries based on referral answers from servers.

Domain Name System (DNS) data is divided into manageable sets of data called *zones*. And each of these sets is a specific zone type.

This example depicts a simple subnet with a Domain Name System (DNS) server connected directly to the Internet.

Domain Name System domain setup

Domain Name System (DNS) domain setup requires domain name registration to prevent others from using your domain name.

DNS allows you to serve names and addresses on an intranet, or internal network. It also allows you to serve names and addresses to the rest of the world through the Internet. If you want to set up domains on the Internet, you are required to register a domain name.

If you are setting up an intranet, you are not required to register a domain name for internal use. Whether to register an intranet name depends on whether you want to ensure that no one else can ever use the name on the Internet, independent of your internal use. Registering a name that you are going to use internally ensures that you will never have a conflict if you later want to use the domain name externally.

Domain registration can be performed by direct contact with an authorized domain name registrar, or through some Internet Service Providers (ISPs). Some ISPs offer a service to submit domain name registration requests on your behalf. The Internet Network Information Center (InterNIC) maintains a directory of all domain name registrars that are authorized by the Internet Corporation for Assigned Names and Numbers (ICANN).

This example depicts a simple subnet with a Domain Name System (DNS) server connected directly to the Internet.

Related information

Internet Network Information Center (InterNIC)

Dynamic updates

i5/OS Domain Name System (DNS) that is based on BIND 9 supports dynamic updates. Outside sources, such as Dynamic Host Configuration Protocol (DHCP), can send updates to the DNS server. In addition, you can also use DNS client tools, such as Dynamic Update Utility (NSUPDATE), to perform dynamic updates.

DHCP is a TCP/IP standard that uses a central server to manage IP addresses and other configuration details for an entire network. A DHCP server responds to requests from clients, dynamically assigning properties to them. DHCP allows you to define network host configuration parameters at a central location and automate the configuration of hosts. It is often used to assign temporary IP addresses to clients for networks that contain more clients than the number of IP addresses available.

In the past, all DNS data was stored in static databases. All DNS resource records had to be created and maintained by the administrator. But, DNS servers that are based on BIND 8, or later, can be configured to accept requests from other sources to update zone data dynamically.

You can configure your DHCP server to send update requests to the DNS server each time it assigns a new address to a host. This automated process reduces DNS server administration in rapidly growing or changing TCP/IP networks, and in networks where hosts change locations frequently. When a client

using DHCP receives an IP address, that data is immediately sent to the DNS server. Using this method, DNS can continue to successfully resolve queries for hosts, even when their IP addresses change.

You can configure DHCP to update address mapping (A for IPv4 or AAAA for IPv6) records, reverse-lookup pointer (PTR) records, or both on behalf of a client. The address mapping record (A or AAAA) maps a machine's host name to its IP address. The PTR record maps a machine's IP address to its host name. When a client's address changes, DHCP can automatically send an update to the DNS server so other hosts in the network can locate the client through DNS queries at the client's new IP address. For each record that is updated dynamically, an associated Text (TXT) record is written to identify that the record was written by DHCP.

Note: If you set DHCP to update only PTR records, you must configure DNS to allow updates from clients so that every client can update its A record if the client uses IPv4 address, or update its AAAA record if the client uses IPv6 address. Not all DHCP clients support making their own A or AAAA record update requests. Consult the documentation for your client platform before choosing this method.

Dynamic zones are secured by creating a list of authorized sources that are allowed to send updates. You can define authorized sources using individual IP addresses, whole subnets, packets that have been signed using a shared secret key (called a *Transaction Signature*, or TSIG), or any combination of those methods. DNS verifies that incoming request packets are coming from an authorized source before updating the resource records.

Dynamic updates can be performed between DNS and DHCP on a single System i platform, between different System i platforms, or between a System i platform and other systems that are capable of dynamic updates.

Note: The dynamic Update DNS (QTOBUPDT) API is required on servers that are sending dynamic updates to DNS. It is installed automatically with i5/OS Option 31, DNS. However, in BIND 9, the NSUPDATE command is the preferred method to make the updates on the System i platform.

Related concepts

Dynamic Host Configuration Protocol

Related tasks

"Configuring Domain Name System to receive dynamic updates" on page 29
Domain Name System (DNS) servers running BIND 9 can be configured to accept requests from other sources to update zone data dynamically. This topic provides instructions for configuring the allow-update option so DNS can receive dynamic updates.

Configuring the DHCP to send dynamic updates to DNS

Related reference

"Example: Domain Name System and Dynamic Host Configuration Protocol on the same System i" on page 18
This example depicts Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) on the same System i platform.

"Domain Name System resource records" on page 9
Resource records are used to store data about domain names and IP addresses. You can use the Resource record lookup table to look into the resource records supported for the i5/OS operating system.

QTOBUPT

"BIND 9 features" on page 8
BIND 9 is similar to BIND 8; however, it provides several features to enhance performance of your Domain Name System (DNS) server, such as views.

BIND 9 features

BIND 9 is similar to BIND 8; however, it provides several features to enhance performance of your Domain Name System (DNS) server, such as views.

Views on a single i5/OS DNS server

The *view* statement allows a single DNS instance to answer a query differently depending on where the query comes from, such as the Internet or an intranet.

One practical application of the view feature is to split DNS setups without having to run multiple DNS servers. For example, in a single DNS server, you can define a view to answer queries from an internal network, while define another view to answer queries from external network.

New client commands

The following client commands enhance the management capability of your DNS server:

Dynamic Update Utility (NSUPDATE)
> The Dynamic Update Utility (NSUPDATE) command is used to submit Dynamic DNS Update requests as defined in Request for Comments (RFC) 2136 to a DNS server. This allows resource records to be added or removed from a zone while the DNS server is running. Thus, you do not need to update records by manually editing the zone file. A single update request can contain requests to add or remove multiple resource records, but the resource records that are dynamically added or removed with the NSUPDATE command should be in the same zone.
>
> **Note:** Do not manually edit zones that are under dynamic control by using the NSUPDATE command or through a DHCP server. Manual edits might conflict with dynamic updates and cause data to be lost.

Start DIG Query (DIG)
> Domain Information Groper (DIG) is a more powerful query tool, compared with the Name Server Lookup (NSLOOKUP) command, that you can use to retrieve information from a DNS server or test the response of a DNS server. The NSLOOKUP command is deprecated and is only provided for compatibility with earlier versions. You can use DIG to verify that a DNS server is responding correctly before you configure your system to use it. You can also retrieve DNS information about hosts, domains, and other DNS servers by using DIG.
>
> You can use the Start DIG Query (STRDIGQRY) command or its alias DIG to start the Domain Information Groper tool.

Start HOST Query (HOST)
> The Start HOST Query (HOST) command is used for DNS lookups. You can use it to convert domain names to IP addresses (either IPv4 or IPv6) and vice versa.

Remote Name Daemon Control (RNDC)

The Remote Name Daemon Control (RNDC) command is a powerful utility that allows a system administrator to control the operation of a name server. It reads a configuration file, called rndc.conf, to determine how to contact the name server and to determine what algorithm and key it should use. If no rndc.conf file is found, then, by default, an rndc-key._KID file that is created during installation is used, which automatically grants access through the loopback interface.

IPv6 support

BIND 9 supports name-to-address and address-to-name lookups in all currently defined forms of IPv6. For forward lookups, BIND 9 supports both AAAA and A6 records, but A6 records are now deprecated.

For IPv6 reverse lookups, it supports the traditional "nibble" format used in the ip6.arpa domain, as well as the older, deprecated ip6.int domain.

Journal files

Journal files are used to hold dynamic updates for a zone. It is automatically created when the first dynamic update from a client is received, and does not disappear. This is a binary file and should not be edited.

With the journal file, when a server is restarted after a shutdown or crash, it replays the journal file to incorporate into the zone any updates that took place after the last zone dump. Journal files are also used to store updates for the incremental zone transfers (IXFR) method.

DNS for i5/OS has been redesigned to use BIND 9. To run BIND 9 DNS on your system, your system must meet certain software requirements.

Related concepts

"Domain Name System requirements" on page 26
Consider these software requirements to run Domain Name System (DNS) on your System i platform.

"Dynamic updates" on page 6
i5/OS Domain Name System (DNS) that is based on BIND 9 supports dynamic updates. Outside sources, such as Dynamic Host Configuration Protocol (DHCP), can send updates to the DNS server. In addition, you can also use DNS client tools, such as Dynamic Update Utility (NSUPDATE), to perform dynamic updates.

"What's new for V6R1" on page 1
Read about new or significantly changed information for the Domain Name System (DNS) topic collection.

Related reference

"Example: Splitting DNS over firewall by setting up two DNS servers on the same System i" on page 20
This example depicts a Domain Name System (DNS) server that operates over a firewall to protect internal data from the Internet, while allowing internal users to access data on the Internet. This configuration accomplishes this protection by setting up two DNS servers on the same System i platform.

"Planning security measures" on page 25
Domain Name System (DNS) provides security options to limit outside access to your server.

Domain Name System resource records

Resource records are used to store data about domain names and IP addresses. You can use the Resource record lookup table to look into the resource records supported for the i5/OS operating system.

A DNS zone database is made up of a collection of resource records. Each resource record specifies information about a particular object. For example, address mapping (A) records map a host name to an IP address, and reverse-lookup pointer (PTR) records map an IP address to a host name. The server uses these records to answer queries for hosts in its zone. For more information, use the table to view DNS resource records.

Note: The entries in the resource record lookup table might be added or removed according to the change of the BIND document. Also, this is not a comprehensive list of all resource records listed in BIND.

Table 1. Resource record lookup table

Resource record	Abbreviation	Description
Address Mapping records	A	The A record specifies the IP address of this host. A records are used to resolve a query for the IP address of a specific domain name. This record type is defined in Request for Comments (RFC) 1035.
Andrew File System Database records	AFSDB	The AFSDB record specifies the AFS or DCE address of the object. AFSDB records are used like A records to map a domain name to its AFSDB address; or to map from the domain name of a cell to authenticated name servers for that cell. This record type is defined in RFC 1183.
Canonical Name records	CNAME	The CNAME record specifies the actual domain name of this object. When DNS queries an aliased name and finds a CNAME record pointing to the canonical name, it then queries that canonical domain name. This record type is defined in RFC 1035.
Host Information records	HINFO	The HINFO record specifies general information about a host. Standard CPU and operating system names are defined in the Assigned Numbers RFC 1700. However, use of the standard numbers is not required. This record type is defined in RFC 1035.
Integrated Services Digital Network records	ISDN	The ISDN record specifies the address of this object. This record maps a host name to the ISDN address. They are used only in ISDN networks. This record type is defined in RFC 1183.
IP Version 6 Address records	AAAA	The AAAA record specifies the 128-bit IPv6 address of a host. AAAA records, which are similar to A records, are used to resolve queries for the IPv6 address of a specific domain name. This record type is defined in RFC 1886.
Location records	LOC	The LOC record specifies the physical location of network components. These records can be used by applications to evaluate network efficiency or map the physical network. This record type is defined in RFC 1876.

Table 1. Resource record lookup table (continued)

Resource record	Abbreviation	Description
Mail Exchanger records	MX	The MX records defines a mail exchanger host for mail sent to this domain. These records are used by Simple Mail Transfer Protocol (SMTP) to locate hosts that processes or forwards mail for this domain, along with preference values for each mail exchanger host. Each mail exchanger host must have a corresponding host address (A) records in a valid zone. This record type is defined in RFC 1035.
Mail Group records	MG	The MG records specifies the mail group domain name. This record type is defined in RFC 1035.
Mailbox records	MB	The MB records specifies the host domain name which contains the mailbox for this object. Mail sent to the domain is directed to the host specified in the MB record. This record type is defined in RFC 1035.
Mailbox Information records	MINFO	The MINFO records specifies the mailbox that should receive messages or errors for this object. The MINFO record is more commonly used for mailing lists than for a single mailbox. This record type is defined in RFC 1035.
Mailbox Rename records	MR	The MR records specifies a new domain name for a mailbox. Use the MR record as a forwarding entry for a user who has moved to a different mailbox. This record type is defined in RFC 1035.
Name Server records	NS	The NS record specifies an authoritative name server for this host. This record type is defined in RFC 1035.
Network Service Access Protocol records	NSAP	The NSAP record specifies the address of a NSAP resource. NSAP records are used to map domain names to NSAP addresses. This record type is defined in RFC 1706.
Public Key records	KEY	The KEY record specifies a public key that is associated with a DNS name. The key can be for a zone, a user, or a host. This record type is defined in RFC 2065.
Responsible Person records	RP	The RP record specifies the internet mail address and description of the person responsible for this zone or host. This record type is defined in RFC 1183.

Table 1. Resource record lookup table (continued)

Resource record	Abbreviation	Description
Reverse-lookup Pointer records	PTR	The PTR record specifies the domain name of a host for which you want a PTR record defined. PTR records allow a host name lookup, given an IP address. This record type is defined in RFC 1035.
Route Through records	RT	The RT record specifies a host domain name that can act as a forwarder of IP packets for this host. This record type is defined in RFC 1183.
Services records	SRV	The SRV record specifies the hosts that support the defined services in the record. This record type is defined in RFC 2782.
Start of Authority records	SOA	The SOA record specifies that this server is authoritative for this zone. An authoritative server is the best source for data within a zone. The SOA record contains general information about the zone and reload rules for secondary servers. There can be only one SOA record per zone. This record type is defined in RFC 1035.
Text records	TXT	The TXT record specifies multiple strings of text, up to 255 characters long each, to be associated with a domain name. TXT records can be used along with responsible person (RP) records to provide information about who is responsible for a zone. This record type is defined in RFC 1035. TXT records are used by i5/OS DHCP for dynamic updates. The DHCP server writes an associated TXT record for each PTR and an A record update that is done by the DHCP server. DHCP records have a prefix of AS400DHCP.
Well-Known Services records	WKS	The WKS record specifies the well-known services supported by the object. Most commonly, WKS records indicate whether tcp or udp or both protocols are supported for this address. This record type is defined in RFC 1035.
X.400 Address Mapping records	PX	The PX records is a pointer to X.400/RFC 822 mapping information. This record type is defined in RFC 1664.

Table 1. Resource record lookup table (continued)

Resource record	Abbreviation	Description
X25 Address Mapping records	X25	The X25 record specifies the address of an X25 resource. This record maps a host name to the PSDN address. They are used only in X25 networks. This record type is defined in RFC 1183.

Related concepts

"Mail and Mail Exchanger records"
Domain Name System (DNS) supports advanced mail routing through the use of Mail and Mail Exchanger (MX) records.

Related reference

"Example: Single Domain Name System server for an intranet" on page 14
This example depicts a simple subnet with a Domain Name System (DNS) server for internal use.

"Understanding zones" on page 3
Domain Name System (DNS) data is divided into manageable sets of data called *zones*. And each of these sets is a specific zone type.

Mail and Mail Exchanger records

Domain Name System (DNS) supports advanced mail routing through the use of Mail and Mail Exchanger (MX) records.

Mail and MX records are used by mail routing programs, such as Simple Mail Transfer Protocol (SMTP). The lookup table in DNS resource records contains the types of mail records that i5/OS DNS supports.

DNS includes information for sending electronic mail by using mail exchanger information. If the network is using DNS, an SMTP application does not deliver mail addressed to host TEST.IBM.COM by opening a TCP connection to TEST.IBM.COM. SMTP first queries the DNS server to find out which host servers can be used to deliver the message.

Deliver mail to a specific address

DNS servers use resource records that are known as *mail exchanger* (MX) records. MX records map a domain or host name to a preference value and host name. MX records are generally used to designate that one host is used to process mail for another host. The records are also used to designate another host to deliver mail to, if the first host cannot be reached. In other words, they allow a mail that is addressed to one host to be delivered to a different host.

Multiple MX resource records might exist for the same domain or host name. When multiple MX records exist for the same domain or host, the preference (or priority) value of each record determines the order in which they are tried. The lowest preference value corresponds to the most preferred record, which is tried first. When the most preferred host cannot be reached, the sending mail application tries to contact the next, less preferred MX host. The domain administrator, or the creator of the MX record, sets the preference value.

A DNS server can respond with an empty list of MX resource records when the name is in the DNS server's authority but has no MX assigned to it. When this occurs, the sending mail application might try to establish a connection with the destination host directly.

Note: Using a wildcard (example: `*.mycompany.com`) in MX records for a domain is not suggested.

Example: MX record for a host

In the following example, the system, by preference, delivers mail for fsc5.test.ibm.com to the host itself. If the host cannot be reached, the system might deliver the mail to psfred.test.ibm.com or to mvs.test.ibm.com (if psfred.test.ibm.com also cannot be reached). This is an example of what these MX records will look like:

```
fsc5.test.ibm.com   IN MX 0 fsc5.test.ibm.com
                    IN MX 2 psfred.test.ibm.com
                    IN MX 4 mvs.test.ibm.com
```

Related reference

"Domain Name System resource records" on page 9
Resource records are used to store data about domain names and IP addresses. You can use the Resource record lookup table to look into the resource records supported for the i5/OS operating system.

Examples: Domain Name System

You can use these examples to understand how to use Domain Name System (DNS) in your network.

DNS is a distributed database system for managing host names and their associated IP addresses. The following examples help to explain how DNS works, and how you can use it in your network. The examples describe the setup and reasons it will be used. They also link to related concepts that you might find useful to understand the pictures.

Example: Single Domain Name System server for an intranet

This example depicts a simple subnet with a Domain Name System (DNS) server for internal use.

The following figure depicts DNS running on a System i platform for an internal network. This single DNS server instance is set up to listen for queries on all interface IP addresses. The system is a primary name server for the mycompany.com zone.
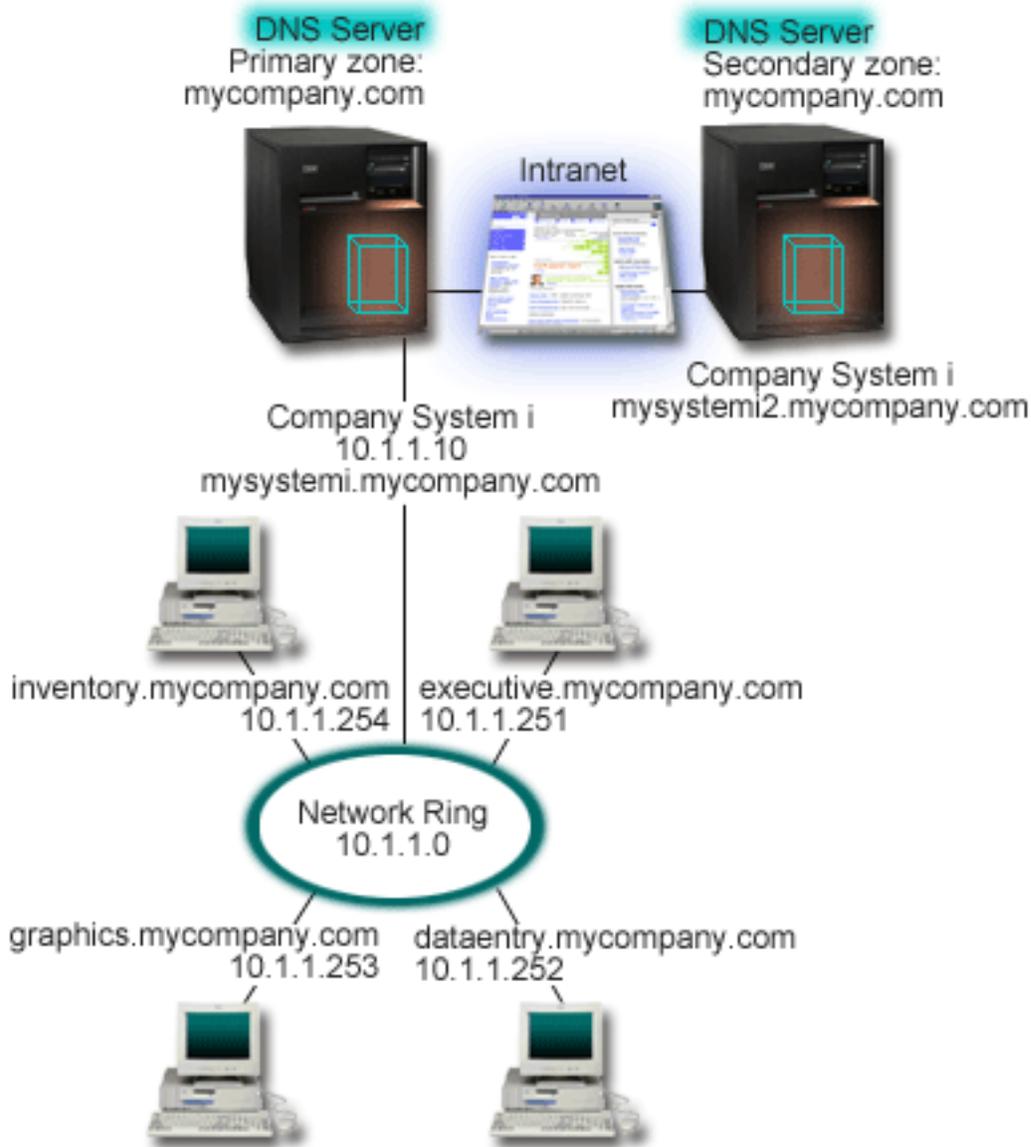
Figure 2. Single DNS server for an intranet

Each host in the zone has an IP address and a domain name. The administrator must manually define the hosts in the DNS zone data by creating resource records. Address mapping records (A for IPv4 or AAAA for IPv6) map the name of a machine to its associated IP address. This allows other hosts on the network to query the DNS server to find the IP address assigned to a particular host name. Reverse-lookup pointer (PTR) records map the IP address of a machine to its associated name. This allows other hosts on the network to query the DNS server to find the host name that corresponds to an IP address.

In addition to A, AAAA, and PTR records, DNS supports many other resource records that might be required, depending on what other TCP/IP-based applications you are running on your intranet. For example, if you are running internal e-mail systems, you might need to add mail exchanger (MX) records so that SMTP can query DNS to find out which systems are running the mail servers.

If this small network were part of a larger intranet, it might be necessary to define internal root servers.

Secondary servers

Secondary servers load zone data from the authoritative server. Secondary servers obtain zone data by doing zone transfers from the authoritative server. When a secondary name server starts, it requests all data for the specified domain from the primary name server. A secondary name server requests updated data from the primary server either because it receives notification from the primary name server (if the NOTIFY function is being used) or because it queries the primary name server and determines that the data has changed. In the figure above, the mysystemi server is part of an intranet. Another system, mysystemi2, has been configured to act as a secondary DNS server for the mycompany.com zone. The secondary server can be used to balance the demand on servers and also to provide a backup in case the primary server goes down. It is a good practice to have at least one secondary server for every zone.

Related reference

"Domain Name System resource records" on page 9
Resource records are used to store data about domain names and IP addresses. You can use the Resource record lookup table to look into the resource records supported for the i5/OS operating system.

"Understanding zones" on page 3
Domain Name System (DNS) data is divided into manageable sets of data called *zones*. And each of these sets is a specific zone type.

"Example: Single Domain Name System server with Internet access"
This example depicts a simple subnet with a Domain Name System (DNS) server connected directly to the Internet.

Example: Single Domain Name System server with Internet access

This example depicts a simple subnet with a Domain Name System (DNS) server connected directly to the Internet.

The following figure depicts the same example network from the single DNS server for intranet example, but now the company has added a connection to the Internet. In this example, the company is able to access the Internet, but the firewall is configured to block Internet traffic into the network.
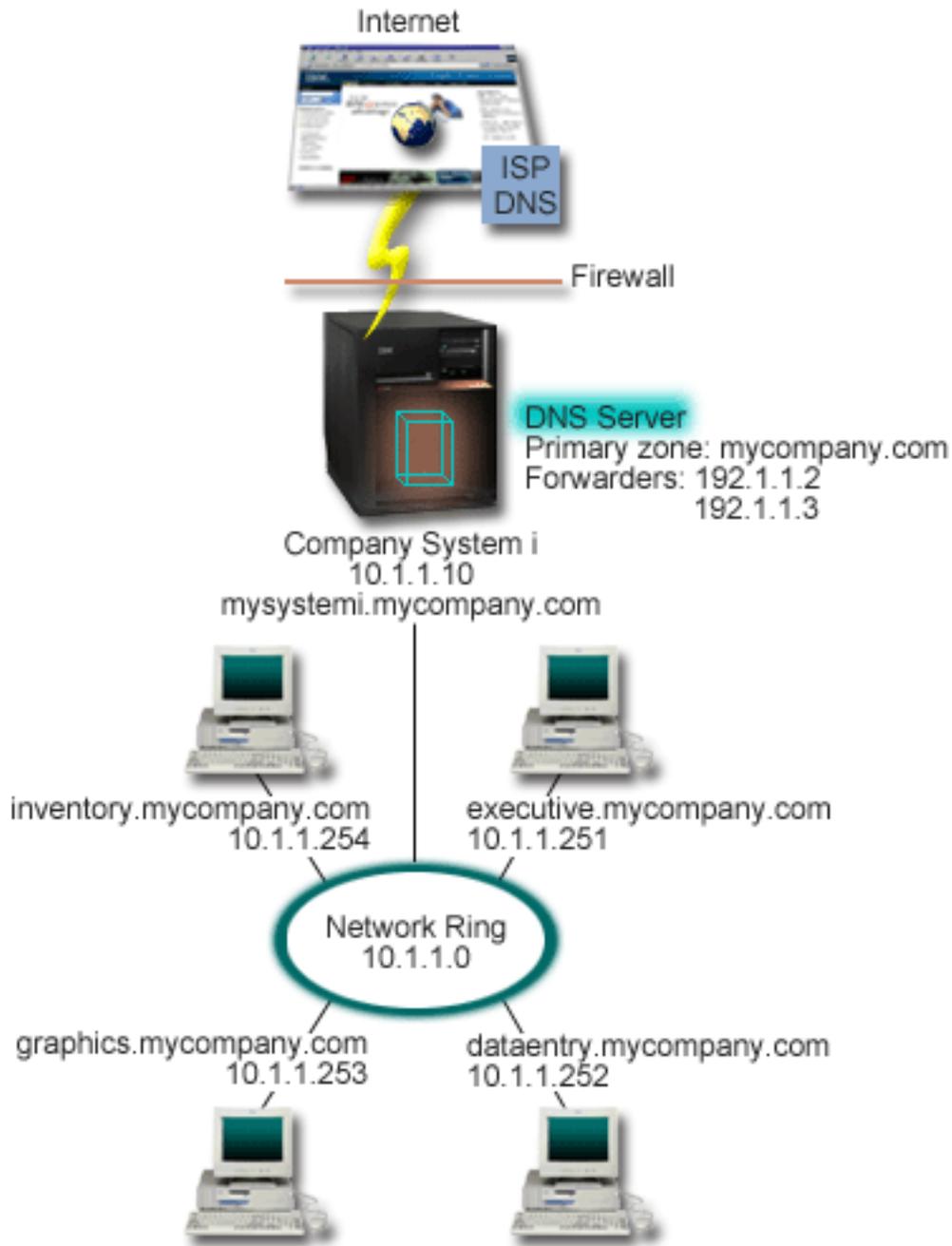
Figure 3. Single DNS server with Internet access

To resolve Internet addresses, you need to do at least one of the following tasks:

* Define Internet root servers

  You can load the default Internet root servers automatically, but you might need to update the list. These servers can help to resolve addresses outside of your own zone. For instructions for obtaining the current Internet root servers, see Accessing external Domain Name System data.

* Enable forwarding

  You can set up forwarding to pass queries for zones outside of mycompany.com to external DNS servers, such as DNS servers run by your Internet service provider (ISP). If you want to enable

searching by both forwarding and root servers, you need to set the `forward` option to **first**. The server first tries forwarding and then queries the root servers only if forwarding fails to resolve the query.

The following configuration changes might also be required:

- Assign unrestricted IP addresses

  In the example above, 10.x.x.x addresses are shown. However, these are restricted addresses and cannot be used outside of an intranet. They are shown below for example purposes, but your own IP addresses is determined by your ISP and other networking factors.

- Register your domain name

  If you are visible to the Internet and have not already registered, you need to register a domain name.

- Establish a firewall

  It is not suggested that you allow your DNS to be directly connected to the Internet. You need to configure a firewall or take other precautions to secure your System i platform.

  Related concepts

  "Domain Name System domain setup" on page 6
  Domain Name System (DNS) domain setup requires domain name registration to prevent others from using your domain name.

  System i and Internet security

  "Understanding Domain Name System queries" on page 4
  Domain Name System (DNS) clients use DNS servers to resolve queries. The queries might come directly from the client or from an application running on the client.

  Related reference

  "Example: Single Domain Name System server for an intranet" on page 14
  This example depicts a simple subnet with a Domain Name System (DNS) server for internal use.

Example: Domain Name System and Dynamic Host Configuration Protocol on the same System i

This example depicts Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) on the same System i platform.

The configuration can be used to update DNS zone data dynamically when DHCP assigns IP addresses to hosts.

The following figure depicts a small subnet network with one System i platform that acts as a DHCP and DNS server to four clients. In this work environment, suppose that the inventory, data entry, and executive clients create documents with graphics from the graphics file server. They connect to the graphics file server by a network drive to its host name.
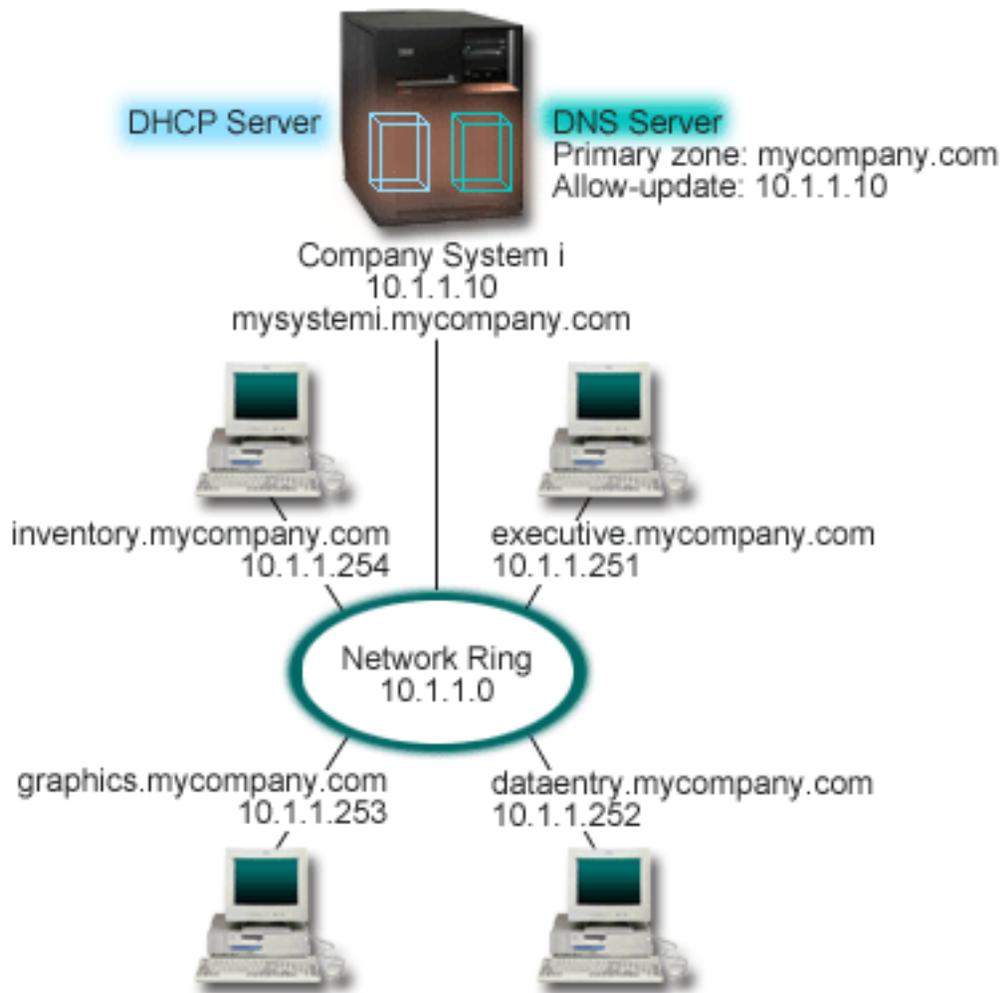
Figure 4. DNS and DHCP on the same System i platform

Previous versions of DHCP and DNS were independent of each other. If DHCP assigned a new IP address to a client, the DNS records had to be manually updated by the administrator. In this example, if the graphics file server's IP address changes because it is assigned by DHCP, then its dependent clients will be unable to map a network drive to its host name because the DNS records will contain the file server's previous IP address.

With the i5/OS DNS server based on BIND 9, you can configure your DNS zone to accept dynamic updates to DNS records in conjunction with intermittent address changes through DHCP. For example, when the graphics file server renews its lease and is assigned an IP address of 10.1.1.250 by the DHCP server, the associated DNS records are updated dynamically. This allows the other clients to query the DNS server for the graphics file server by their host names without interruption.

To configure a DNS zone to accept dynamic updates, complete the following tasks:

• Identify the dynamic zone

  You cannot manually update a dynamic zone while the server is running. Doing so might cause interference with incoming dynamic updates. Manual updates can be made when the server is stopped, but you will lose any dynamic updates sent while the server is down. For this reason, you might want to configure a separate dynamic zone to minimize the need for manual updates. See Determining domain structure for more information about configuring your zones to use the dynamic update function.

- Configure the allow-update option

  Any zone with the allow-update option configured is considered a dynamic zone. The allow-update option is set on a per-zone basis. To accept dynamic updates, the allow-update option must be enabled for this zone. For this example, the mycompany.com zone has allow-update data, but other zones defined on the server can be configured to be static or dynamic.

- Configure DHCP to send dynamic updates

  You must authorize your DHCP server to update the DNS records for the IP addresses it has distributed.

- Configure secondary server update preferences

  To keep secondary servers current, you can configure DNS to use the NOTIFY function to send a message to secondary servers for the mycompany.com zone when zone data changes. You should also configure incremental zone transfers (IXFR), which enables IXFR-enabled secondary servers to track and load only the updated zone data, instead of the entire zone.

If you run DNS and DHCP on different servers, there are some additional configuration requirements for the DHCP server.

Related concepts

"Dynamic updates" on page 6
i5/OS Domain Name System (DNS) that is based on BIND 9 supports dynamic updates. Outside sources, such as Dynamic Host Configuration Protocol (DHCP), can send updates to the DNS server. In addition, you can also use DNS client tools, such as Dynamic Update Utility (NSUPDATE), to perform dynamic updates.

Related tasks

Configuring the DHCP to send dynamic updates to DNS

Related reference

Example: DNS and DHCP on different System i platforms

Example: Splitting DNS over firewall by setting up two DNS servers on the same System i

This example depicts a Domain Name System (DNS) server that operates over a firewall to protect internal data from the Internet, while allowing internal users to access data on the Internet. This configuration accomplishes this protection by setting up two DNS servers on the same System i platform.

The following figure depicts a simple subnet network that uses a firewall for security. Suppose that the company has an internal network with reserved IP space and an external section of a network that is available to the public. The company wants its internal clients to be able to resolve external host names and to exchange mail with people on the outside. The company also wants its internal resolvers to have access to certain internal-only zones that are not available at all outside of the internal network. However, they do not want any outside resolvers to be able to access the internal network.

With i5/OS DNS based on BIND 9, you can use two ways to accomplish this. The first way is that the company sets up two DNS server instances on the same System i platform, one for the intranet and another for everything in its public domain, which is described in this example. Another way is to use the view function that is provided in BIND 9, which is described in the example about splitting DNS over firewall by using a view.
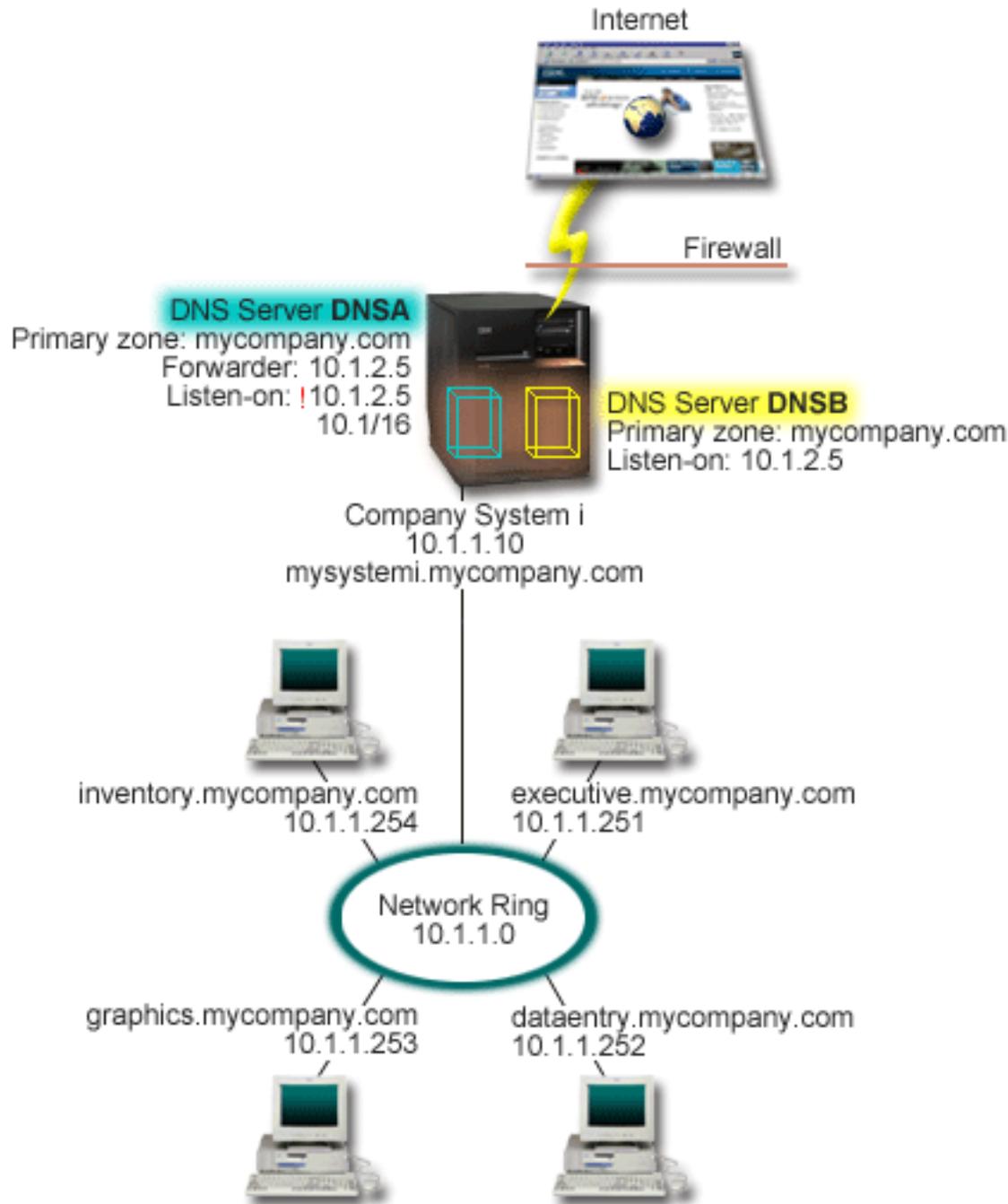
Figure 5. Splitting DNS over a firewall by setting up two DNS servers on the same System i

The external server, DNSB, is configured with the primary zone mycompany.com. This zone data includes only the resource records that are intended to be part of the public domain. The internal server, DNSA, is configured with the primary zone mycompany.com, but the zone data defined on DNSA contains intranet resource records. The forwarder option is defined as 10.1.2.5. This forces DNSA to forward queries it cannot resolve to the DNSB server.

If you are concerned about the integrity of your firewall or other security threats, you have the option of using the listen-on option to help protect internal data. To do this, you can configure the internal server to only allow queries to the internal mycompany.com zone from internal hosts. In order for all this to

work correctly, internal clients need to be configured to query only the DNSA server. You need to consider the following configuration settings to split DNS:

- Listen-on

  In other DNS examples, only one DNS server is on a System i platform. It is set to listen on all interface IP addresses. Whenever you have multiple DNS servers on a System i platform, you must define the interface IP addresses that each one listens on. Two DNS servers cannot listen on the same address. In this case, assume that all queries that come in from the firewall are sent in on 10.1.2.5. These queries should be sent to the external server. Therefore, DNSB is configured to listen on 10.1.2.5. The internal server, DNSA, is configured to accept queries from anything on the 10.1.x.x interface IP addresses except 10.1.2.5. To effectively exclude this address, the address match list must have the excluded address listed before the included address prefix.

- Address match list order

  The first element in the address match list that a given address matches is used. For example, to allow all addresses on the 10.1.x.x network except 10.1.2.5, the ACL elements must be in the order (!10.1.2.5; 10.1/16). In this case, the address 10.1.2.5 is compared to the first element and is immediately denied.

  If the elements are reversed (10.1/16; !10.1.2.5), the IP address 10.1.2.5 is allowed access because the server compares it to the first element that matches, and allows it without checking the rest of the rules.

  Related reference

  "BIND 9 features" on page 8
  BIND 9 is similar to BIND 8; however, it provides several features to enhance performance of your Domain Name System (DNS) server, such as views.

  "Example: Splitting DNS over firewall by using view"
  This example depicts a Domain Name System (DNS) server that operates over a firewall to protect internal data from the Internet, while allowing internal users to access data on the Internet by using the *view* feature that BIND 9 provides.

Example: Splitting DNS over firewall by using view

This example depicts a Domain Name System (DNS) server that operates over a firewall to protect internal data from the Internet, while allowing internal users to access data on the Internet by using the *view* feature that BIND 9 provides.

The following figure depicts a simple subnet network that uses a firewall for security. Suppose that the company has an internal network with reserved IP space and an external section of a network that is available to the public. The company wants its internal clients to be able to resolve external host names and to exchange mail with people outside the network. The company also wants its internal resolvers to have access to certain internal-only zones that are not available outside of the internal network. However, the company does not want any outside resolvers to be able to access the internal network.

With i5/OS DNS based on BIND 9, you can use two ways to accomplish this. The way described in this example is that you can configure the DNS server with two different views to listen on various queries, one for the intranet and another for everything in its public domain. Another way is to set up two DNS server instances on the same System i platform, which is described in the example about splitting DNS over a firewall by using two DNS servers.

Internet

DNS Server
DNSC

Firewall

View "internal"
Primary zone: mycompany.com
match-clients:!10.1.2.5
10.1/16

View "external"
Primary zone: mycompany.com
match-clients: 10.1.2.5

Company System i
10.1.1.10
mysystemi.mycompany.com

inventory.mycompany.com
10.1.1.254

executive.mycompany.com
10.1.1.251

Network Ring
10.1.1.0

graphics.mycompany.com
10.1.1.253

dataentry.mycompany.com
10.1.1.252

Figure 6. Splitting DNS over a firewall by using view

The DNS server, DNSC, defines two views, called *external* and *internal*. The *external* view is configured with a primary zone mycompany.com that includes only the resource records that are intended to be part of the public domain, while the *internal* view is configured with a primary zone mycompany.com that contains intranet resource records.

If you are concerned about the integrity of your firewall or other security threats, you have the option of using the match-clients substatement to help protect internal data. To do this, you can configure the internal view to only allow queries to the internal mycompany.com zone from internal hosts. You need to consider the following configuration settings to set up split DNS:

- Match-clients

The match-clients in a view statement takes an address match list as an argument. Only a query's IP address that matches the address match list can see the configuration values defined in the enclosing view. If a query's IP address matches multiple match-clients entries in various view statements, the first view statement is the one that applies. In this case, assume that all queries that come from the firewall are sent in 10.1.2.5. These queries should be handled by the zone data in the external view. Therefore, 10.1.2.5 is set to be the match-clients of the external view. The internal view is configured to accept queries from anything on the 10.1.x.x interface IP addresses except 10.1.2.5. To effectively exclude this address, the address match list must have the excluded address listed before the included address prefix.

- Address match list order

  The first element in the address match list that a given address matches is used. For example, to allow all addresses on the 10.1.x.x network except 10.1.2.5, the ACL elements must be in the order (!10.1.2.5; 10.1/16). In this case, the address 10.1.2.5 is compared to the first element and is immediately denied.

  If the elements are reversed (10.1/16; !10.1.2.5), the IP address 10.1.2.5 is allowed access because the server compares it to the first element that matches, and allow it without checking the rest of the rules.

  Related reference

  "Example: Splitting DNS over firewall by setting up two DNS servers on the same System i" on page 20

  This example depicts a Domain Name System (DNS) server that operates over a firewall to protect internal data from the Internet, while allowing internal users to access data on the Internet. This configuration accomplishes this protection by setting up two DNS servers on the same System i platform.

Planning for Domain Name System

Domain Name System (DNS) offers a variety of solutions. Before you configure DNS, it is important to plan how it works within your network. Subjects, such as network structure, performance, and security, should be assessed.

Determining Domain Name System authorities

There are special authorization requirements for the Domain Name System (DNS) administrator. You should also consider security implications of authorization.

When you set up DNS, you should take security precautions to protect your configuration. You need to establish which users are authorized to make changes to the configuration.

A minimum level of authority is required to allow your administrator to configure and administer DNS. Granting all object access ensures that the administrator is capable of performing DNS administrative tasks. It is suggested that users who configure DNS have security officer access with all object (*ALLOBJ) authority. Use System i Navigator to authorize users. If you need more information, refer to the Granting authority to the DNS administrator topic in the DNS online help.

Note: If an administrator's profile does not have full authority, specific access and authority to all DNS directories and related configuration files must be granted.

Related reference

"Maintaining Domain Name System configuration files" on page 35
You can use i5/OS DNS to create and manage DNS server instances on your System i platform. The configuration files for DNS are managed by System i Navigator. You must not manually edit the files. Always use System i Navigator to create, change, or delete DNS configuration files.

Determining domain structure

If you are setting up a domain for the first time, you should plan for demand and maintenance before creating zones.

It is important to determine how you divide your domain or subdomains into zones, how to best serve network demand, access to the Internet, and how to negotiate firewalls. These factors can be complex and must be dealt with case-by-case. Refer to authoritative sources such as the O'Reilly DNS and BIND book for in-depth guidelines.

If you configure a Domain Name System (DNS) zone as a dynamic zone, you cannot make manual changes to zone data while the server is running. Doing so might cause interference with incoming dynamic updates. If it is necessary to make manual updates, stop the server, make the changes, and then restart the server. Dynamic updates sent to a stopped DNS server will never be completed. For this reason, you might want to configure a dynamic zone and a static zone separately. You can do this by creating entirely separate zones, or by defining a new subdomain, such as dynamic.mycompany.com, for those clients that will be maintained dynamically.

i5/OS DNS provides a graphical interface for configuring your systems. In some cases, the interface uses terminology or concepts that might be represented differently in other sources. If you refer to other information sources when you are planning for your DNS configuration, it might be helpful to remember the following items:

- All zones and objects defined on a System i platform are organized within the folders Forward Lookup Zones and Reverse Lookup Zones. Forward lookup zones are the zones that are used to map domain names to IP addresses, such as A and AAAA records. The reverse lookup zones are the zones that are used to map IP addresses to domain names, such as PTR records.
- i5/OS DNS refers to *primary zones* and *secondary zones*.
- The interface uses *subzones*, which some sources refer to as *subdomains*. A child zone is a subzone for which you have delegated responsibility to one or more name servers.

Planning security measures

Domain Name System (DNS) provides security options to limit outside access to your server.

Address match lists

DNS uses address match lists to allow or deny outside entities access to certain DNS functions. These lists can include specific IP addresses, a subnet (using an IP prefix), or using Transaction Signature (TSIG) keys. You can define a list of entities to which you want to allow or deny access in an address match list. If you want to be able to reuse an address match list, you can save the list as an access control list (ACL). Then whenever you need to provide the list, you can call the ACL and the entire list will be loaded.

Address match list item order

The first item in an address match list that a given address matches is used. For example, to allow all addresses on the 10.1.1.x network except 10.1.1.5, the match list items must be in the order (!10.1.1.5; 10.1.1/24). In this case, the address 10.1.1.5 will be compared to the first item and will immediately be denied.

If the elements are reversed (10.1.1/24; !10.1.1.5), the IP address 10.1.1.5 will be allowed access because the server will compare it to the first item, which matches, and allow it without checking the rest of the rules.

Access control options

DNS allows you to set limitations such as who can send dynamic updates to the server, query data, and request zone transfers. You can use ACLs to restrict access to the server for the following options:

allow-update
> In order for your DNS server to accept dynamic updates from any outside sources, you must enable the allow-update option.

allow-query

> Specifies which hosts are allowed to query this server. If not specified, the default is to allow queries from all hosts.

allow-transfer

> Specifies which hosts are allowed to receive zone transfers from the server. If not specified, the default is to allow transfers from all hosts.

allow-recursion

> Specifies which hosts are allowed to make recursive queries through this server. If not specified, the default is to allow recursive queries from all hosts.

blackhole

> Specifies a list of addresses that the server does not accept queries from or use to resolve a query. Queries from these addresses will not be responded to.

Securing your DNS server is essential. In addition to the security considerations in this topic, DNS security and System i security are covered in a variety of sources including the System i platform and the Internet topic collection. The book *DNS and BIND* also covers security related to DNS.

> **Related concepts**

> System i and Internet security

> **Related reference**

> "BIND 9 features" on page 8
> BIND 9 is similar to BIND 8; however, it provides several features to enhance performance of your Domain Name System (DNS) server, such as views.

Domain Name System requirements

Consider these software requirements to run Domain Name System (DNS) on your System i platform.

The DNS feature, Option 31, cannot be installed automatically with the operating system. You must specifically select DNS for installation. The DNS server added for i5/OS is based on the industry-standard DNS implementation known as BIND 9. Previous OS/400® DNS services were based on BIND 8.2.5, and are still available in i5/OS.

After DNS is installed, you are required to migrate and configure the DNS server from BIND 4 or 8 to BIND 9. You must also have i5/OS PASE installed, which is Option 33 of i5/OS. After i5/OS PASE is installed, System i Navigator automatically handles configuring the current BIND implementation.

If you want to configure a Dynamic Host Configuration Protocol (DHCP) server on a different platform to send updates to this DNS server, Option 31 must be installed on that DHCP server as well. The DHCP server uses programming interfaces provided by Option 31 to perform dynamic updates.

> **Related concepts**

> i5/OS PASE

> "Configuring Domain Name System" on page 27
> You can use System i Navigator to configure name servers and to resolve queries outside of your domain.

> **Related reference**

> "BIND 9 features" on page 8
> BIND 9 is similar to BIND 8; however, it provides several features to enhance performance of your Domain Name System (DNS) server, such as views.

Determining if Domain Name System is installed

To determine if Domain Name System (DNS) is installed, follow these steps.

1. At the command line, type `GO LICPGM` and press Enter.
2. Type 10 (Display installed licensed programs) and press Enter.
3. Page down to **5761SS1 Domain Name System** (Option 31). If DNS is installed successfully, the Installed Status is `*COMPATIBLE`, as shown here:

   ```
   LicPgm     Installed Status    Description
   5761SS1    *COMPATIBLE         Domain Name System
   ```
4. Press F3 to exit the display.

Installing Domain Name System

To install Domain Name System (DNS), follow these steps .

1. At the command line, type `GO LICPGM` and press Enter.
2. Type 11 (Install licensed programs) and press Enter.
3. Type 1 (Install) in the **Option** field next to `Domain Name System` and press Enter.
4. Press Enter again to confirm the installation.

Configuring Domain Name System

You can use System i Navigator to configure name servers and to resolve queries outside of your domain.

Before you work with your Domain Name System (DNS) configuration, see DNS system requirements to install the necessary DNS components.

> **Related concepts**
> "Domain Name System requirements" on page 26
> Consider these software requirements to run Domain Name System (DNS) on your System i platform.

Accessing Domain Name System in System i Navigator

These instructions guide you to the DNS configuration interface in System i Navigator.

About this task

If you are using i5/OS PASE, you will be able to configure DNS servers based on BIND 9.

If you are configuring DNS for the first time, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. Right-click **DNS** and select **New Configuration**.

   > **Related concepts**
   > Getting to know System i Navigator

Configuring name servers

Domain Name System (DNS) allows you to create multiple name server instances. This topic provides instructions for configuring a name server.

About this task

i5/OS DNS based on BIND 9 supports multiple name server instances. The following tasks guide you through the process of creating a single name server instance, including its properties and zones.

Results

If you want to create multiple instances, repeat these procedures until all instances you want have been created. You can specify independent properties, such as debug levels and autostart values, for each name server instance. When you create a new instance, separate configuration files are created.

Related reference

"Maintaining Domain Name System configuration files" on page 35
You can use i5/OS DNS to create and manage DNS server instances on your System i platform. The configuration files for DNS are managed by System i Navigator. You must not manually edit the files. Always use System i Navigator to create, change, or delete DNS configuration files.

Creating a name server instance

The New Domain Name System (DNS) Configuration wizard can guide you through the process of defining a DNS server instance.

About this task

To start the **New DNS Configuration** wizard, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the left pane, right-click **DNS** and select **New Name Server**.
3. Follow the wizard's instructions to complete the configuration process.

Results

The wizard requires the following input:

DNS server name:
Specify a name for your DNS server. It can be up to 5 characters long and must begin with an alphabetic character (A-Z). If you create multiple servers, each must have a unique name. This name is referred to as the DNS server instance name in other areas of the system.

Listen-on IP addresses:
Two DNS servers cannot listen on the same IP address. The default setting is to listen on all IP addresses. If you are creating additional server instances, they cannot be configured to listen on all IP addresses. Otherwise, they cannot be run at the same time. You must specify the IP addresses for each server.

Root servers:
You might load the list of default Internet root servers or specify your own root servers, such as internal root servers for an intranet.

> **Note:** You should only consider loading the default Internet root servers if you have access to the Internet and expect your DNS to be able to fully resolve Internet names.

Server startup:
You can specify whether the server should autostart when TCP/IP is started. When you operate multiple servers, individual instances can be started and ended independently of each other.

Editing Domain Name System server properties

After you create a name server, you can edit properties such as allow-update and debug levels. These options apply only to the server instance you change.

About this task

To edit the properties of the Domain Name System (DNS) server instance, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.

2. In the right pane, right-click *your DNS server* and select **Configuration**.

3. In the DNS Configuration window, right-click **DNS Server** and select **Properties**.

4. Edit the corresponding properties you'd like to.

Configuring zones on a name server

After you configure a Domain Name System (DNS) server instance, you need to configure the zones for the name server.

About this task

To configure zones on your server, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.

2. In the right pane, right-click *your DNS server* and select **Configuration**.

3. In the DNS Configuration window, select the zone type that you want to create by right-clicking either the **Forward Lookup Zone** or the **Reverse Lookup Zone** folder.

4. Follow the wizard's instructions to complete the creation process.

   Related concepts

   "Accessing external Domain Name System data" on page 31
   When you create Domain Name System (DNS) zone data, your server will be able to resolve queries to that zone.

   Related tasks

   "Configuring Domain Name System to receive dynamic updates"
   Domain Name System (DNS) servers running BIND 9 can be configured to accept requests from other sources to update zone data dynamically. This topic provides instructions for configuring the allow-update option so DNS can receive dynamic updates.

   "Importing Domain Name System files" on page 30
   Domain Name System (DNS) can import existing zone data files. Follow these time-saving procedures for creating a new zone from an existing configuration file.

   Related reference

   "Understanding zones" on page 3
   Domain Name System (DNS) data is divided into manageable sets of data called *zones*. And each of these sets is a specific zone type.

Configuring views on a name server

One of the features that BIND 9 offers is the *view* statement, which allows a single Domain Name System (DNS) instance to answer a query differently depending on where the query is coming from, such as the Internet or an intranet. One practical application of view is to split DNS setups without having to run multiple DNS servers.

About this task

To configure views on your server, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.

2. In the right pane, right-click *your DNS server* and select **Configuration**.

3. In the DNS Configuration window, right-click **Views** and select **New View**.

4. Follow the wizard's instructions to complete the creation process.

Configuring Domain Name System to receive dynamic updates

Domain Name System (DNS) servers running BIND 9 can be configured to accept requests from other sources to update zone data dynamically. This topic provides instructions for configuring the allow-update option so DNS can receive dynamic updates.

About this task

When creating dynamic zones, you should consider your network structure. If parts of your domain still requires manual updates, you might want to consider setting up separate static and dynamic zones. If you need to make manual updates to a dynamic zone, you must stop the dynamic zone server and restart it after you have completed the updates. Stopping the server forces it to update the zone database with all dynamic updates that have been made since the server first loaded its zone data from the zone database. If you do not stop the server, you will lose any manual updates to the zone database because they will be overwritten by the running server. However, stopping the server to make manual updates means you might miss dynamic updates that are sent while the server is down.

DNS indicates that a zone is dynamic when objects are defined in the allow-update statement. To configure the allow-update option, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the right pane, right-click *your DNS server* and select **Configuration**.
3. In the DNS Configuration window, expand **Forward Lookup Zone** or **Reverse Lookup Zone**.
4. Right-click the primary zone that you want to edit and select **Properties**.
5. In the Primary Zone Properties page, click the **Options** tab.
6. On the Options page, expand **Access Control** → **allow-update**.
7. DNS uses an address match list to verify authorized updates. To add an object to the address match list, select an address match list item type and click **Add**. You can add an IP Address, IP Prefix, Access Control List, or Key.
8. When you have finished updating the address match list, click **OK** to close the Options page.

   Related tasks

   "Configuring zones on a name server" on page 29
   After you configure a Domain Name System (DNS) server instance, you need to configure the zones for the name server.
   Configuring the DHCP to send dynamic updates to DNS

Importing Domain Name System files

Domain Name System (DNS) can import existing zone data files. Follow these time-saving procedures for creating a new zone from an existing configuration file.

About this task

You can create a primary zone by importing a zone data file that is a valid zone configuration file based on BIND syntax. The file should be located in an Integrated file system directory. When imported, DNS verifies that it is a valid zone data file and adds it to the named.conf file for the specified server instance.

To import a zone file, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the right pane, double-click the DNS server instance into which you want to import the zone.
3. In the left pane of the DNS Configuration window, right-click **DNS server** and select **Import Zone**.
4. Follow the wizard's instructions to import the primary zone.

   Related tasks

   "Configuring zones on a name server" on page 29
   After you configure a Domain Name System (DNS) server instance, you need to configure the zones for the name server.

Record validation

The Import domain data function reads and validates each record of the file that is being imported.

About this task

After the Import domain data function has finished, any records in error can be examined individually on the Other Records property page of the imported zone.

Notes:

1. Importing a large primary domain might take several minutes.
2. The import domain data function does not support the $include directive. Import domain data's validity checking process identifies lines that contain the $include directive as lines in error.

Accessing external Domain Name System data

When you create Domain Name System (DNS) zone data, your server will be able to resolve queries to that zone.

Root servers are critical to the function of a DNS server that is directly connected to the Internet or a large intranet. DNS servers must use root servers to answer queries about hosts other than those that are contained in their own domain files.

To reach out for more information, a DNS server has to know where to look. On the Internet, the first place that a DNS server looks is the root servers. The root servers direct a DNS server toward other servers in the hierarchy until an answer is found, or it is determined that there is no answer.

The default root servers list for System i Navigator

You should use Internet root servers only if you have an Internet connection and you want to resolve names on the Internet if they are not resolved on your DNS server. A default list of Internet root servers is supplied in System i Navigator. The list is current when System i Navigator is released. You can verify that the default list is current by comparing it to the list on the InterNIC site. Update your configuration's root server list to keep it current.

Getting Internet root server addresses

The top-level root server's addresses change from time to time, and it is the DNS administrator's responsibility to keep them current. InterNIC maintains a current list of Internet root server addresses. To obtain a current list of Internet root servers, follow these steps:

1. Log on the InterNIC server by using File Transfer Protocol (FTP) in the anonymity method: `FTP.INTERNIC.NET` or `RS.INTERNIC.NET`
2. Download this file: `/domain/named.root`
3. Store the file in the following directory path: `/QOpenSys/QIBM/ProdData/OS400/DNS/ROOT.FILE`

A DNS server behind a firewall might have no root servers defined. In this case, the DNS server can resolve queries only from entries that exist in its own primary domain database files, or its cache. It might forward off-site queries to the firewall DNS. In this case, the firewall DNS server acts as a forwarder.

Intranet root servers

If your DNS server is part of a large intranet, you might have internal root servers. If your DNS server will not be accessing the Internet, you do not need to load the default Internet servers. However, you should add your internal root servers so that your DNS server can resolve internal addresses outside of its domain.

Related tasks

After you configure a Domain Name System (DNS) server instance, you need to configure the zones for the name server.

Managing Domain Name System

Managing a Domain Name System (DNS) server includes verifying that the DNS function is working, monitoring performance, and maintaining DNS data and files.

Verifying the Domain Name System function is working

The domain information groper (DIG) tool can help you collect information from and test response of a Domain Name System (DNS) server. You can use DIG to verify if a DNS server is working correctly.

About this task

Request the host name that is associated with the loopback IP address (127.0.0.1). It should respond with the host name (localhost). You can also query specific names that are defined in the server instance that you are trying to verify. This confirms that the specific server instance you are testing is functioning correctly.

To verify DNS function with DIG, follow these steps:

1. At the command line, type `DIG HOSTNAME('127.0.0.1') REVERSE(*YES)`.

   This information should display, including the loopback host name:

   ```
   ;; global options: printcmd
   ;; Got answer:
   ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id:865
   ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:1

   ;; QUESTION SECTION:
   ;1.0.0.127.in-addr.arpa.          IN    PTR

   ;; ANSWER SECTION:
   1.0.0.127.in-addr.arpa.  86400   IN    PTR  localhost.

   ;; AUTHORITY SECTION:
   0.0.127.in-addr.arpa.    86400   IN    NS   ISA2LP05.RCHLAND.IBM.COM.

   ;; ADDITIONAL SECTION:
   ISA2LP05.RCHLAND.IBM.COM. 38694   IN    A   9.5.176.194

   ;; Query time: 552 msec
   ;; SERVER: 9.5.176.194#53(9.5.176.194)
   ;; WHEN: Thu May 31 21:38:12 2007
   ;; MSG SIZE rcvd: 117
   ```

   The DNS server is responding correctly if it returns the loopback host name: **localhost**.
2. Press Enter to quit the session.

Results

Note: If you need help using DIG, type `?DIG` and press Enter.

Managing security keys

Security keys allow you to limit access to your Domain Name System (DNS) data.

About this task

There are two types of keys related to DNS, which are DNS keys and dynamic update keys. They each play a different role in securing your DNS configuration. The following descriptions explain how each relates to your DNS server.

Managing Domain Name System keys
The Domain Name System (DNS) keys are keys defined for BIND and used by the DNS server as part of the verification of an incoming update.

About this task

You can configure a key and assign it a name. Then, when you want to protect a DNS object, such as a dynamic zone, you can specify the key in the Address Match List.

To manage DNS keys, follow these steps:
1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the right pane, right-click the DNS server instance that you want to manage and select **Configuration**.
3. In the DNS Configuration window, select **File** → **Manage Keys**.

Results

In the Manage Keys window, you can perform the corresponding management tasks.

Managing dynamic update keys
Dynamic update keys are used for securing dynamic updates by the Dynamic Host Configuration Protocol (DHCP) server.

About this task

These keys must be present when Domain Name System (DNS) and DHCP are on the same System i platform. If DHCP is on a different System i platform, you must distribute the same dynamic update key files to each remote System i platform that needs them to send dynamic updates to the authoritative servers. You can distribute them through FTP, e-mail, and so forth.

To manage dynamic update keys, follow these steps:
1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. Right-click **DNS** and select **Manage Dynamic Update Keys**.

Results

You can then perform the corresponding management tasks in the Manage Dynamic Update Keys window.

Accessing Domain Name System server statistics

Database dump and statistics tools can help you review and manage server performance.

Domain Name System (DNS) provides several diagnostic tools. They can be used to monitor performance of your server.

Related reference
"Maintaining Domain Name System configuration files" on page 35
You can use i5/OS DNS to create and manage DNS server instances on your System i platform. The

configuration files for DNS are managed by System i Navigator. You must not manually edit the files. Always use System i Navigator to create, change, or delete DNS configuration files.

Accessing server statistics

The server statistics summarize the number of queries and responses the server received since the last time the server restarted or reloaded its database.

About this task

Domain Name System (DNS) allows you to view the statistics for a server instance. Information is continually appended to this file until you delete the file. This information might be useful in evaluating how much traffic the server receives, and in tracking down problems. More information about server statistics is available in the DNS online help topic Understanding DNS server statistics.

To access server statistics, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the right pane, right-click *your DNS server* and select **Configuration**.
3. In the DNS configuration window, select **View** → **Server Statistics**.

Results

You can also use the Remote Name Daemon Control (RNDC) command to display the server statistics information in the named.stats file. The corresponding command is as follows.

```
RNDC RNDCCMD('stats')
```

Accessing an active server database

The active server database contains zone and host information, including some zone properties, such as start of authority (SOA) information, and through host properties, such as mail exchanger (MX) information, which might be useful in tracking down problems.

About this task

Domain Name System (DNS) allows you to view a dump of the authoritative data, cache data, and hints data for a server instance. The dump includes the information from all of the server's primary and secondary zones (forward and reverse mapping zones), as well as information that the server has obtained from queries.

You can view the active server database dump using System i Navigator. If you need to save a copy of the files, the database dump file name is named_dump.db in your i5/OS directory path: /QIBM/UserData/OS400/DNS/<server instance>/, where *<server instance>* is the name of the DNS server instance. More information about the active server database is available in the DNS online help topic Understanding the DNS server database dump.

To access the active server database dump, follow these steps:

1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the right pane, right-click *your DNS server* and select **Configuration**.
3. In the DNS configuration window, select **View** → **Active Server Database**.

Results

You can also use the Remote Name Daemon Control (RNDC) command to display the active server database information in the named_dump.db file. The corresponding command is as follows.

```
RNDC RNDCCMD('dumpdb -all')
```

Maintaining Domain Name System configuration files

You can use i5/OS DNS to create and manage DNS server instances on your System i platform. The configuration files for DNS are managed by System i Navigator. You must not manually edit the files. Always use System i Navigator to create, change, or delete DNS configuration files.

DNS configuration files are stored in the integrated file system paths listed below.

Note: The file structure below applies to DNS running on BIND 9.

In the following table, files are listed in the hierarchy of paths shown. Files with a save icon 💾 should be backed up to protect data. Files with a delete icon ✖ should be deleted on a regular basis.

Name	Icon	Description
/QIBM/UserData/OS400/DNS/		Starting point directory for DNS.
/QIBM/UserData/OS400/DNS/ *<instance-n>***/**		Starting point directory for a DNS instance.
ATTRIBUTES	💾	DNS uses this file to determine which BIND version you are using.
BOOT.AS400BIND4	💾	BIND 4.9.3 server configuration and policies file that is converted to the BIND 8 named.conf file for this instance. This file is created if you migrate a BIND 4.9.3 server to BIND 9. It serves as a backup for migration, and can be deleted when the BIND 9 server is working properly.
named.ca	💾	List of root servers for this server instance.
named.conf	💾	This file contains configuration data. It tells the server what specific zones it is managing, where the zone files are, which zones can be dynamically updated, where its forwarding servers are, and other option settings.
named_dump.db	✖	Server data dump created for the active server database.
named.memstats	✖	Server memory statistics (if configured in named.conf).
named.pid		Holds Process ID of running server. This file is created each time the DNS server is started. It is used for the Database, Statistics, and Update server functions. Do not delete or edit this file.
named.random		Server generated entropy file.
named.recursing	✖	Servers queries that are recursive (if requested by System i Navigator).
named.run	✖	Default debug log (if requested). It can roll over as named.run.0, named.run.1, and so on.
named.stats	✖	Server statistics.

Name	Icon	Description
<primary-zone-n>.db	💾	It is the primary zone file for a particular domain on this server. The file contains all of the resource records for this zone. Each zone has a separate .db file.
<primary-zone-n>.jnl	💾	Journal file that holds dynamic updates for a zone. It is created when the first dynamic update is received. When a server is restarted after a shutdown or crash, it replays the journal file to incorporate into the zone any updates that took place after the last zone dump. This file is also used for incremental zone transfers (IXFR). These log files do not disappear. This is a binary file and should not be edited.
db.*<secondary-zone-n>*	💾	Secondary zone file for a particular domain on this server. Contains all of the resource records for this zone. This file is used to initially load the secondary server at startup if the primary server is unreachable. Each zone has a separate .db file.
/QIBM/UserData/OS400/DNS/_DYN/		Directory that holds files required for dynamic updates.
<key_id-n>._KEY		.Symlink to DNSSEC key with the *<key_id-n>* key. It always points to the last K*<key_id-n>*.*+aaa+nnnnn*.key key that is created.
<key_id-x>._DUK. *<zone-a>*	💾	Dynamic update key required to initiate a dynamic update request to *<zone-a>* using the *<key_id-x>* key.
<key_id-x>._KID	💾	File containing a key statement for the key_id named *<key_id-x>*
<key_id-y>._DUK. *<zone-a>*	💾	Dynamic update key required to initiate a dynamic update request to *<zone-a>* using the *<key_id-y>* key.
<key_id-y>._DUK. *<zone-b>*	💾	Dynamic update key required to initiate a dynamic update request to *<zone-b>* using the *<key_id-y>* key.
<key_id-y>._KID	💾	File containing a key statement for the key_id named *<key_id-y>*
rndc-confgen.random.*nnnnnn*	✖	Entropy files for various commands that require them. The *nnnnn* part is the job number of the job that created the file. These are only left behind if the command cancels for some reason and does not clean up.

Related concepts

"Determining Domain Name System authorities" on page 24

There are special authorization requirements for the Domain Name System (DNS) administrator. You should also consider security implications of authorization.

Database dump and statistics tools can help you review and manage server performance.

Related tasks

Domain Name System (DNS) allows you to create multiple name server instances. This topic provides
instructions for configuring a name server.

Advanced Domain Name System features

This topic explains how experienced administrators can use Domain Name System (DNS) advanced
features to manage a DNS server more easily.

DNS in System i Navigator provides an interface with advanced features for configuring and managing
your DNS server. The following tasks are provided as shortcuts for administrators who are familiar with
the i5/OS graphical interface. They provide fast methods for changing server status and attributes for
multiple instances simultaneously.

Related tasks

The Domain Name System (DNS) debug function can provide information that can help you
determine and correct DNS server problems.

Starting or stopping Domain Name System servers

If Domain Name System (DNS) in the System i Navigator interface does not allow you to start or stop
multiple server instances simultaneously, you can use the character-based interface to change these
settings for multiple instances simultaneously.

About this task

To use the character-based interface to start all DNS server instances at once, type STRTCPSVR
SERVER(*DNS) DNSSVR(*ALL) at the command line. To stop all DNS servers at once, type ENDTCPSVR
SERVER(*DNS) DNSSVR(*ALL) at the command line.

Changing debug values

It is useful to change the debug level for administrators who have large zones and do not want the large
amount of debug data collected when the server is first starting up and loading all of the zone data.

About this task

Domain Name System (DNS) in the System i Navigator interface does not allow you to change the debug
level while the server is running. However, you can use the character-based interface to change the debug
level while the server is running. To change the debug level using the character-based interface, follow
these steps, replacing *nnnnn* in the command with the name of the server instance:

1. At the command line, type ADDLIBLE QDNS and press Enter.
2. Change the debug level:

   - To turn debugging on or to increase the debug level by 1, type RNDC RNDCCMD('trace') and press
     Enter.
   - To turn debugging off, type RNDC RNDCCMD('notrace') and press Enter.

Troubleshooting Domain Name System

Domain Name System (DNS) logging and debugging settings can help you resolve problems with your
DNS server.

DNS operates much the same as other TCP/IP functions and applications. Like SMTP or FTP
applications, DNS jobs run under the QSYSWRK subsystem and produce job logs under the user profile

QTCP with information associated with the DNS job. If a DNS job ends, you can use the job logs to determine the cause. If the DNS server is not returning the expected responses, the job logs might contain information that can help with problem analysis.

The DNS configuration consists of several files with several different types of records in each file. Problems with the DNS server are generally the result of incorrect entries in the DNS configuration files. When a problem occurs, verify that the DNS configuration files contain the entries you expect.

Identifying jobs

If you look in the job log to verify DNS server function (using WRKACTJOB, for example), consider the following naming guidelines:

* If you are running servers based on BIND 9, there will be a separate job for each server instance you are running. The job name is five fixed chars (QTOBD) followed by the instance name. For example, if you have two instances, INST1 and INST2, their job names will be QTOBDINST1 and QTOBDINST2.

Logging Domain Name System server messages

Domain Name System (DNS) provides numerous logging options that can be adjusted when you are trying to find the source of a problem. Logging provides flexibility by offering various severity levels, message categories, and output files so that you can fine-tune logging to help you find problems.

BIND 9 offers several logging options. You can specify what types of messages are logged, where each message type is sent, and what severity of each message type to log. In general, the default logging settings are suitable, but if you want to change them, it is suggested that you refer to other sources of BIND 9 documentation for information about logging.

Logging channels

The DNS server can log messages to different output channels. Channels specify where logging data is sent. You can select the following channel types:

* **File channels**

    Messages logged to file channels are sent to a file. The default file channels are i5os_debug and i5os_QPRINT. By default, debug messages are logged to the i5os_debug channel, which is the named.run file, but you can specify to send other message categories to this file as well. Message categories logged to i5os_QPRINT are sent to a QPRINT spooled file for user profile QTCP. You can create your own file channels in addition to the default channels provided.

* **Syslog channels**

    Messages logged to this channel are sent to the server's job log. The default syslog channel is i5os_joblog. Logging messages routed to this channel are sent to the job log of the DNS server instance.

* **Null channels**

    All messages logged to the null channel are discarded. The default null channel is i5os_null. You can route categories to the null channel if you do not want the messages to appear in any log file.

Message categories

Messages are grouped into categories. You can specify what message categories should be logged to each channel. The categories are as follows:

client   Processing of client requests.

config   Configuration file parsing and processing.

database
    Messages relating to the databases that are used internally by the DNS server to store zone and cache data.

default

Definitions of the logging options for those categories where no specific configuration has been defined.

delegation-only

Delegation only. It logs queries that have been forced to NXDOMAIN as the result of a delegation-only zone or a delegation-only in a hint or stub zone declaration.

dispatch

Dispatching of incoming packets to the server modules where they are to be processed.

dnssec

DNS Security Extensions (DNSSEC) and Transaction Signature (TSIG) protocol processing.

general

The catch-all category that is used for those things that are not classified into any other categories.

lame-servers

Lame servers that are misconfigurations in remote servers, discovered by BIND 9 when trying to query those servers during resolution.

network

Network operations.

notify The NOTIFY protocol.

resolver

DNS resolution, such as the recursive lookups, that is performed on behalf of clients by a caching name server.

security

Approval and denial of requests.

xfer-in Zone transfers that the server is receiving.

xfer-out

Zone transfers that the server is sending.

unmatched

Messages that are named was unable to determine the class of or for which there was no matching view. A one-line summary is also logged to the client category. This category is best sent to a file or stderr. By default, it is sent to the null channel.

update

Dynamic updates.

update-security

Approval and denial of update requests. Queries specify where queries should be logged. At startup, specifying the category queries enables query logging unless the querylog option is specified.

The query log entry reports the client's IP address and port number, the query name, class, and type. It also reports whether the Recursion Desired flag was set (+ if set, - if not set), EDNS was in use (E), or if the query was signed (S).

Log files can become large and can be deleted on a regular basis. All contents in the DNS log file are cleared when the DNS server is stopped and started.

Message severity

Channels allow you to filter by message severity. For each channel, you can specify the severity level for which messages are logged. The following severity levels are available:

- Critical
- Error
- Warning
- Notice
- Info
- Debug (specify debug level 0-11)
- Dynamic (inherit the server startup debug level)

All messages of the severity you select and any levels above it in the list are logged. For example, if you select Warning, the channel logs Warning, Error, and Critical messages. If you select Debug level, you can specify a value from 0 to 11 for which you want debug messages to be logged.

Changing logging settings

To access logging options, follow these steps:
1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.
2. In the right pane, right-click *your DNS server* and select **Configuration**.
3. In the DNS configuration window, right-click **DNS server** and select **Properties**.
4. In the Server Properties window, select the **Channels** tab to create new file channels or properties of a channel, such as the severity of messages logged to each channel.
5. In the Server Properties window, select the **Logging** tab to specify which message categories are logged to each channel.

Troubleshooting tip about the severity level

The i5os_joblog channel default severity level is set to Error. This setting is used to reduce the volume of informational and warning messages, which can otherwise degrade performance. If you are experiencing problems but the job log is not indicating the source of the problem, you might need to change the severity level. Follow the procedure above to access the Channels page and change the severity level for the i5os_joblog channel to Warning, Notice, or Info so you can view more logging data. After you have resolved the problem, reset the severity level to Error to reduce the number of messages in the job log.

Changing Domain Name System debug settings

The Domain Name System (DNS) debug function can provide information that can help you determine and correct DNS server problems.

About this task

DNS offers 12 levels of debug control. Logging typically provides an easier method of finding problems, but in some cases debugging might be necessary. Under normal conditions, debugging is turned off (value = 0). It is recommended that you first use logging to attempt to correct problems.

Valid debug levels are 0 through 11. Your IBM service representative can help you determine the appropriate debug value for diagnosing your DNS problem. Values of 1 or higher write debug information to the named.run file in your i5/OS directory path: /QIBM/UserData/OS400/DNS/*<server instance>*, where *<server instance>* is the name of the DNS server instance. The named.run file continues to grow as long as the debug level is set to 1 or higher, and the DNS server continues to run. You can also use the Server Properties - Channels page to specify preferences for the maximum size and the number of versions of the named.run file.

To change the debug value for a DNS server instance, follow these steps:
1. In System i Navigator, expand *your system* → **Network** → **Servers** → **DNS**.

2. In the right pane, right-click *your DNS server* and select **Configuration**.
3. In the DNS configuration window, right-click the DNS server and select **Properties**.
4. On the Server Properties - General page, specify the server startup debug level.
5. If the server is running, stop and restart the server.

> **Note:** Changes to the debug level do not take effect while the server is running. The debug level set here will be used the next time the server is fully restarted. If you need to change the debug level while the server is running, see Advanced DNS features.

Related concepts

"Advanced Domain Name System features" on page 37
This topic explains how experienced administrators can use Domain Name System (DNS) advanced features to manage a DNS server more easily.

Related information for Domain Name System

IBM Redbooks publications, Web sites, and other information center topic collections contain information that relates to the Domain Name System (DNS) topic collection. You can view or print any of the PDF files.

IBM Redbooks

AS/400® TCP/IP Autoconfiguration: DNS and DHCP Support (5181 KB)

This Redbooks publication describes the Domain Name System (DNS) server and Dynamic Host Configuration Protocol (DHCP) server support that are included in i5/OS. It can help you install, tailor, configure, and troubleshoot DNS and DHCP support through examples.

Web sites

- *DNS and BIND*, fifth edition. Paul Albitz and Cricket Liu. Published by O'Reilly and Associates,Inc. Sebastopol, California, 2006. ISBN number: 0-59610-057-4.

- The BIND Administrator Reference Manual (in PDF version) from the Internet System Consortium (ISC) Web site.

- The Internet Software Consortium Web site contains news, links, and other resources for BIND.

- The InterNIC site maintains a directory of all domain name registrars that are authorized by the Internet Corporation for Assigned Names and Numbers (ICANN).

- The DNS Resources Directory provides DNS reference material and links to many other DNS resources, including discussion groups. It also provides a listing of DNS related RFCs .

Related reference

"PDF file for Domain Name System" on page 2
You can view and print a PDF file of this information.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This Domain Name System (DNS) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400
i5/OS
IBM
IBM (logo)
OS/400
Redbooks
System i

Adobe, the Adobe logo, PostScript®, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM ®

Printed in USA