IP Addressing: DNS Configuration Guide, Cisco IOS XE Gibraltar 16.11.x

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

CHAPTER 2

Configuring DNS

The Domain Name System (DNS) is a distributed database in which you can map host names to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated host name. The Cisco IOS XE software maintains a cache of host-name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. This cache speeds the process of converting names to addresses.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring DNS

To use DNS, you must have a DNS name server on your network.

Information About DNS

DNS Overview

If your network devices require connectivity with devices in networks for which you do not control name assignment, you can assign device names that uniquely identify your devices within the entire internetwork. The global naming scheme of the Internet, the DNS, accomplishes this task. This service is enabled by default. The following sections summarize DNS concepts and function:

Host Names for Network Devices

Each unique IP address can have an associated host name. DNS uses a hierarchical scheme for establishing host names for network nodes. This allows local control of the segments of the network through a client-server scheme. The DNS system can locate a network device by translating the host name of the device into its associated IP address.

Domains Names for Groups of Networks

IP defines a naming scheme that allows a device to be identified by its location in the IP. This is a hierarchical naming scheme that provides for *domains*. On the Internet, a domain is a portion of the naming hierarchy tree that refers to general groupings of networks based on organization type or geography. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that the IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.cisco.com*.

Name Servers

To keep track of domain names, IP has defined the concept of a *name server*. Name servers are programs that have complete information about their namespace portion of the domain tree and may also contain pointers to other name servers that can be used to lead to information from any other part of the domain tree. Name servers know the parts of the domain tree for which they have complete information. A name server may also store information about other parts of the domain tree. To map domain names to IP addresses, you must first identify the host names, then specify a name server, and enable the DNS service.

Cache

To speed the process of converting names to addresses, the name server maintains a database, called a *cache*, of host-name-to-address mappings for use by the **connect**, **telnet**, and **ping** EXEC commands, and related Telnet support operations. The cache stores the results from previous responses. Upon receiving a client-issued DNS query, it will check this local storage to see if the answer is available locally.

Name Resolvers

Name resolvers are programs that extract information from name servers in response to client requests. Resolvers must be able to access at least one name server. The resolver either uses that name server's information to answer a query directly or pursues the query using referrals to other names servers. A resolver will typically be a system routine that is directly accessible to user programs. Therefore, no protocol is necessary between the resolver and the user program.

Zones

The domain namespace is divided into areas called zones that are points of delegation in the DNS tree. A zone contains all domains from a certain point downward, except those for which other zones are authoritative.

Authoritative Name Servers

A name server is said to be an authority for the parts of the domain tree for which it has complete information. A zone usually has an authoritative name server, often more than one. An authoritative name server has been configured with host table information or has acquired host table information though a zone transfer (the action that occurs when a secondary DNS server starts up and updates itself from the primary server).

DNS Operation

Within an organization, you can have many name servers, but Internet clients can query only those that the root name servers know. The other name servers answer internal queries only.

A name server handles client-issued queries to the DNS server for locally defined hosts within a particular zone as follows:

- An authoritative name server responds to DNS user queries for a domain name that is under its zone of authority by using the permanent and cached entries in its own host table. If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server simply replies that no such information exists..
- A name server that is not configured as the authoritative name server responds to DNS user queries by using information that it has cached from previously received query responses. If no device is configured as the authoritative name server for a zone, queries to the DNS server for locally defined hosts will receive nonauthoritative responses.

Name servers answer DNS queries (forward incoming DNS queries or resolve internally generated DNS queries) according to the forwarding and lookup parameters configured for the specific domain.

DNS Security

An alternating sequence of DNS public key (DNSKEY) RR sets and Delegation Signer (DS) RR sets forms a chain of signed data, with each link in the chain vouching for the next. A DNSKEY RR is used to verify the signature covering a DS RR and allows the DS RR to be authenticated. The DS RR contains a hash of another DNSKEY RR and this new DNSKEY RR is authenticated by matching the hash in the DS RR.

DNS Views

A DNS view is a set of parameters that specify how to handle a DNS query. A DNS view defines the following information:

- Association with a VRF
- Parameters for resolving internally generated DNS queries
- Parameters for forwarding incoming DNS queries
- Internal host table for answering queries or caching DNS responses

Note The maximum number of DNS views and view lists depends on the memory of Cisco device. Configuring a large number of DNS views and view lists uses more device memory, and configuring a large number of views in the view lists uses more device processor time. For optimum performance, configure views and view list members that are required to support your Split DNS query forwarding or query resolution needs.

Restricted View Use Queries from the Associated VRF

A DNS view is always associated with a VRF— the global VRF or a named VRF, so as to limit the view usage in handling DNS queries that arrive on an interface matching a particular VRF:

- A DNS view that is associated with the global VRF can be used only to handle DNS queries that arrive on an interface in the global address space.
- A DNS view that is associated with a named VRF can be used only to handle DNS queries that arrive on an interface that matches the VRF with which the view is associated.

Note Additional restrictions (described in DNS Views) can be placed on a view after it has been defined. Also, a single view can be referenced multiple times, with different restrictions added in each case. However, because the association of a DNS view with a VRF is specified in the DNS view definition, the VRF-specific view-use limitation is a characteristic of the DNS view definition itself and cannot be separated from the view.

Parameters for Resolving Internally Generated DNS Queries

- Domain lookup—Enabling or disabling of DNS lookup to resolve hostnames for internally generated queries.
- Default domain name—Default domain to append to hostnames without a dot.
- Domain search list—List of domain names to try for hostnames without a dot.
- Domain name for multicast lookups—IP address to use for multicast address lookups.
- Domain name servers—List of name servers to use to resolve domain names for internally generated queries.
- Resolver source interface—Source interface to use to resolve domain names for internally generated queries.
- Round-robin rotation of IP addresses—Enabling or disabling of the use of a different IP address associated with the domain name in cache each time hostnames are looked up.

Parameters for Forwarding Incoming DNS Queries

The following parameters define how to forward incoming DNS queries:

- Forwarding of queries—Enabling or disabling of forwarding of incoming DNS queries.

- Forwarder addresses—List of IP addresses to use to forward incoming DNS queries.
- Forwarder source interface—Source interface to use to forward incoming DNS queries.

Sometimes, when a source interface is configured on a device with the split DNS feature to forward DNS queries, the device does not forward the DNS queries through the configured interface. Hence, consider the following points while forwarding the DNS queries using the source interface:

- DNS queries are forwarded to a broadcast address when a forwarding source interface is configured and the DNS forwarder is not configured.
- The source IP address of the forwarded query should be set to the primary IP address of the interface configured, using the **dns forwarding source-interface** *interface* command. If no such configuration exists, then the source IP address of the forwarded DNS query will be the primary IP address of the outgoing interface. DNS forwarding should be done only when the source interface configured for the DNS forwarding is active.
- The source IP address of the DNS query for the DNS resolver functionality is set using the **domain resolver source-interface** *interface-type number* command. If there is no DNS address configured, then queries will be broadcasted to the defined source interface. DNS resolving should be done only when the source interface configured for the DNS resolving is active. See "Specifying a Source Interface to Forward DNS Queries" for the configuration steps.

DNS View Lists

A DNS view list is an ordered list of DNS views in which additional usage restrictions can be specified for any individual member in the list. The scope of these optional usage restrictions is limited to a specific member of a specific DNS view list. When the device must respond to a DNS query, the Cisco IOS software uses a DNS view list to select the DNS view that will be used to handle a DNS query.

Note The maximum number of DNS views and view lists supported is not specifically limited but is dependent on the amount of memory on the Cisco device. Configuring a larger number of DNS views and view lists uses more device memory, and configuring a larger number of views in the view lists uses more device processor time. For optimum performance, configure no more views and view list members than needed to support your Split DNS query forwarding or query resolution needs.

Order in Which to Check the Members of a DNS View List

When a DNS view list is used to select a DNS view for handling a given DNS query, the Cisco IOS software checks each member of the view list--in the order specified by the list--and selects the first view list member whose restrictions permit the view to be used with the query that needs to be handled.

Usage Restrictions Defined for a DNS View in the View List

A DNS view list member can be configured with usage restrictions defined using access control lists (ACLs) that specify rules for selecting that view list member based on the query hostname or the query source host IP address. The two types of ACLs supported by the Split DNS view list definition are described in "DNS Name Groups".

> **Note** Multiple DNS view lists can be defined so that, for example, a given DNS view can be associated with different restrictions in each list. Also, different DNS view lists can include different DNS views.

Selection of the DNS View List

When the device that is acting as the DNS caching name server needs to respond to a DNS query, the Cisco IOS software uses a DNS view list to determine which DNS view can be used to handle the query:

- If the device is responding to an incoming query that arrives on an interface for which a DNS view list is configured, the interface-specific DNS view list is used.
- If the device is responding to an incoming query that arrives on an interface for which no specific DNS view list is configured, the default DNS view list is used.

If the device is responding to an internally generated query, no DNS view list is used to select a view; the global DNS view is used to handle the query.

The assignment of a DNS view list as the default or to an interface is described in "DNS View Groups".

Selection of a DNS View List Member

The view list members are compared, each in turn, to the characteristics of the DNS query that the device is responding to:

1. If the query is from a different VRF than the view, the view cannot be used to address the query, so the view-selection process moves on to the next member of the view list.
2. The specification of additional view-use restrictions is an optional setting for any view list member.

If the query list does not specify additional restrictions on the view, the view will be used to address the query, so the view-selection process is finished.

If the view list does specify additional restrictions on the view, the query is compared to those restrictions:

- If the query characteristics fail any view-use restriction, the view cannot be used to address the query, so the view-selection process moves on to the next member of the view list.
- If the query characteristics pass all the view-use restrictions, the view will be used to address the query. The view-selection process is finished.
- If the view-selection process reaches the end of the selected DNS view list without finding a view list member that can handle the query, the device discards the query.

The first DNS view list member that is found to have restrictions that match the query characteristics is used to handle the query.

DNS Name Groups

The Split DNS feature supports two types of ACLs that can be used to restrict the use of a DNS view. A DNS name list or a standard IP ACL (or both) can be applied to a DNS view list member to specify view-use restrictions in addition to the VRF-specific restriction that is a part of the view definition itself.

> **Note** In this context, the term “group” is used to refer to the specification of a DNS name list or a standard IP ACL as a usage restriction on a view list member.

DNS View Usage Restrictions Based on the Query Hostname

A DNS name list is a named set of hostname pattern-matching rules, with each rule specifying the type of action to be performed if a query hostname matches the text string pattern in the rule. In order for a query hostname to match a name list, the hostname must match a rule that explicitly permits a matching pattern but the hostname cannot match any rules that explicitly deny a matching pattern.

DNS View Usage Restrictions Based on the Query Source IP Address

A standard IP ACL is a numbered or named set of host IP address-matching rules, with each rule specifying the type of action to be performed if an IP address matches the text string pattern in the rule. The Split DNS feature supports the use of a standard ACL as a view-use restriction based on the query source IP address. In order for a source IP address to match a name list, the IP address must match a rule that explicitly permits a matching pattern but the IP address cannot match any rules that explicitly deny a matching pattern.

DNS View Groups

The Split DNS feature provides two ways to specify the DNS view list that the Cisco IOS software is to use to select the DNS view that will be used to handle an incoming DNS query. For a query that arrives on an interface that is configured to use a particular DNS view list, the interface-specific DNS view list is used. Otherwise, the default DNS view list is used.

> **Note** In this context, the term “group” refers to the specification of a DNS view list as an interface-specific DNS view list or the default view list for the device.

Interface-specific View Lists

A DNS view list can be attached to a device interface. When an incoming DNS query arrives on that interface, the Cisco IOS software uses that view list to select a DNS view to use to handle the query.

Default DNS View List

A DNS view list can be configured as the default DNS view list for the device. When an incoming DNS query arrives on an interface that is not configured to use a specific view list, the Cisco IOS software uses the default view list to select the DNS view to use to handle the query.

How to Configure DNS

Mapping Host Names to IP Addresses

Perform this task to associate host names with IP addresses.

A name server is used to keep track of information associated with domain names. A name server can maintain a database of host name-to-address mappings. Each name can map to one or more IP addresses. In order to use this service to map domain names to IP addresses, you must specify a name server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host** *name* [*tcp-port-number*] *address1* [*address2* ... *address8*] [*mx* ns srv]
4. Do one of the following:
   - **ip domain name** *name*
   - **ip domain list** *name*
5. **ip name-server** *server-address1* [*server-address2* ... *server-address6*]
6. **ip domain lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip host** *name* [*tcp-port-number*] *address1* [*address2* ... *address8*] [*mx* ns srv]
	Example:
	`Device(config)# ip host cisco-rtp 192.168.0.148`
	`Device(config)# ip host test mx 1 mx_record`
	`Device(config)# ip host test ns ns_record`
	`Device(config)# ip host test srv 0 0 0 srv_record`
		Defines a static host name-to-address mapping in the host name cache.
		• Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use host names or addresses). Host names and IP addresses can be associated with one another through static or dynamic means.
		• Manually assigning host names to addresses is useful when dynamic mapping is not available.

	Command or Action	Purpose
		• Mail exchanger (mx) identifies the mail server that is responsible for handling e-mails for a given domain name. • Name server (ns) state the authoritative name servers for the given domain. • Service (srv) records specifies the location of a service.

Step 4	Do one of the following:
	• **ip domain name** *name*
	• **ip domain list** *name*
	Example:
	`Device(config)# ip domain name cisco.com`
	Example:
	`Device(config)# ip domain list cisco1.com`
		(Optional) Defines a default domain name that the Cisco IOS XE software will use to complete unqualified host names.
		or
		(Optional) Defines a list of default domain names to complete unqualified host names.
		• You can specify a default domain name that the Cisco IOS XE software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any host name that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.
		Note If there is no domain list, the domain name that you specified with the **ip domain name** global configuration command is used. If there is a domain list, the default domain name is not used. The **ip domain list** command is similar to the **ip domain name** command, except that with the **ip domain list** command you can define a list of domains, each to be tried in turn until the system finds a match.

Step 5	**ip name-server** *server-address1* [*server-address2 ... server-address6*] **Example:** `Device(config)# ip name-server 172.16.1.111 172.16.1.2`	Specifies one or more hosts that supply name information. • Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS.
Step 6	**ip domain lookup** **Example:** `Device(config)# ip domain lookup`	(Optional) Enables DNS-based address translation. • DNS is enabled by default. Use this command if DNS has been disabled.

What to do next

The name lookup system can be statically configured using the commands described in this task. Some other functions in Cisco IOS XE, such as DHCP can dynamically modify the state of the name lookup system. Use the **show hosts** command to display the cached host names and the DNS configuration.

Disabling DNS Queries for ISO CLNS Addresses

Perform this task to disable DNS queries for ISO CLNS addresses.

If your device has both IP and ISO Connectionless Network Service (ISO CLNS) enabled and you want to use ISO CLNS network service access point (NSAP) addresses, you can use the DNS to query these addresses, as documented in RFC 1348. This feature is enabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip domain lookup nsap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ip domain lookup nsap** **Example:** `Device(config)# no ip domain lookup nsap`	Disables DNS queries for ISO CLNS addresses.

Verifying DNS

Perform this task to verify your DNS configuration.

SUMMARY STEPS

1. **enable**
2. **ping** *host*
3. **show hosts**
4. **debug ip domain**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** *host* **Example:** `Device# ping cisco-rtp`	Diagnoses basic network connectivity. • After the DNS configuration is set, you can verify the DNS server by using a hostname to ping or telnet to a device.
Step 3	**show hosts** **Example:** `Device# show hosts`	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of host names and addresses. • After a name is resolved using DNS, use the **show hosts** command to view the cached hostnames and the DNS configuration.
Step 4	**debug ip domain** **Example:** `Device# debug ip domain`	Enables DNS debugging and displays DNS debugging information. • To view more DNS debugging options such as DNS server response debugging and so on, use the question mark (?) online help function.

Defining a DNS View

Perform this task to define a DNS view. A DNS view definition can be used to respond to either an incoming DNS query or an internally generated DNS query.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*}
4. [**no**] **dns trust** *name*
5. [**no**] **domain lookup**
6. Do one of the following:
   - **domain name** *domain-name*
   - **domain list** *domain-name*
7. Do one of the following:
   - **domain name-server** [**vrf** *vrf-name*] *name-server-ip-address*
   - **domain name-server interface** *interface*

8. **domain multicast** *domain-name*
9. [**no**] **dns forwarding**
10. **dns forwarder** [**vrf** *vrf-name*] *forwarder-ip-address*
11. **dns forwarding source-interface** *interface*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*} **Example:** `Device(config)# ip dns view vrf vpn101 user3`	Defines a DNS view and enters DNS view configuration mode.
Step 4	[**no**] **dns trust** *name* **Example:** `Device(cfg-dns-view)# dns trust name`	(Optional) Enables or disables storage of trusted keys in a view and enters DNS view configuration mode. The dns trust key enables the DNS security feature.
Step 5	[**no**] **domain lookup** **Example:** `Device(cfg-dns-view)# domain lookup`	(Optional) Enables or disables DNS-based hostname-to-address translation for internally generated DNS queries handled using the DNS view. **Note** The domain lookup capability is enabled by default.

Step 6	Do one of the following:
	• **domain name** *domain-name*
	• **domain list** *domain-name*
	Example:
	`Device(cfg-dns-view)# domain name example.com`
	Example:
	`Device(cfg-dns-view)# domain list example1.com`
		(Optional) Defines a default domain name to be used by this DNS view to complete unqualified hostnames when addressing DNS queries.
		or
		(Optional) Defines a list of domain names to be used by this DNS view to complete unqualified hostnames when addressing DNS queries.
		• The device attempts to respond to the query using the parameters specified by the selected DNS view. First, the Cisco IOS software looks in the hostname cache associated with the view. If the query can be answered from that information, the device responds to the

	Command or Action	Purpose

		query. Otherwise, because the query cannot be answered using the hostname cache, the device forwards the query using the configured domain name servers.
		• If the device is using this view to handle a DNS query for an unqualified hostname and domain lookup is enabled for the view, the Cisco IOS software appends a domain name (either a domain name from the domain name list or the default domain name) in order to perform any of the following activities:
		  • Looking up the hostname in the name server cache.
		  • Forwarded the query to other name servers (whether to the hosts specified as DNS forwarders in the selected view or to the limited broadcast address).
		• You can specify a single, default domain name, an ordered list of domain names, or both. However, the default domain name is used only if the domain list is empty.

Step 7	Do one of the following:
	• **domain name-server** [**vrf** *vrf-name*] *name-server-ip-address*
	• **domain name-server interface** *interface*
	Example:
	`Device(cfg-dns-view)# domain name-server 192.168.2.124`
	Example:
	`Device(cfg-dns-view)# domain name-server interface FastEthernet0/1`
		(Optional) Defines a list of name servers to be used by this DNS view to resolve internally generated DNS queries. The IP address of the name server can be an IPv4 or IPv6 address, and the IP address can be associated with a Virtual Private Network (VPN) routing and forwarding (VRF) instance.
		or
		(Optional) Defines an interface on which to acquire (through DHCP or PPP interaction on the interface) the IP address of a DNS server to add to the list of DNS name servers to be used by this DNS view to resolve internally generated DNS queries.
		• If both of these commands are configured, DHCP or PPP interaction on the interface causes another IP address to be added to the list.

Step 8	**domain multicast** *domain-name* **Example:** `Device(cfg-dns-view)# domain multicast www.example8.com`	(Optional) Specifies the IP address to use for multicast lookups handled using the DNS view.
Step 9	[**no**] **dns forwarding** **Example:**	(Optional) Enables or disables forwarding of incoming DNS queries handled using the DNS view.

	Command or Action	Purpose
	`Device(cfg-dns-view)# dns forwarding`	**Note** The query forwarding capability is enabled by default.
Step 10	**dns forwarder** [**vrf** *vrf-name*] *forwarder-ip-address* **Example:** `Device(cfg-dns-view)# dns forwarder 192.168.3.240`	Defines a list of name servers to be used by this DNS view to forward incoming DNS queries. • The forwarder IP address can be an IPv4 or IPv6 address. • If no forwarding name servers are defined, then the configured list of domain name servers is used instead. • If no name servers are configured either, then queries are forwarded to the limited broadcast address.
Step 11	**dns forwarding source-interface** *interface* **Example:** `Device(cfg-dns-view)# dns forwarding source-interface FastEthernet0/0`	Defines the interface on which to forward queries when this DNS view is used.
Step 12	**end** **Example:** `Device(cfg-dns-view)# end`	Returns to privileged EXEC mode.

Verifying DNS Views

Perform this task to verify the DNS configuration.

SUMMARY STEPS

1. **enable**
2. **show ip dns view** [**vrf** *vrf-name*] [**default** | *view-name*]
3. **show ip dns server** [**vrf** *vrf-name*] [**default** | *view-name*]
4. **clear ip dns servers**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip dns view** [**vrf** *vrf-name*] [**default** | *view-name*] **Example:** `Device# show ip dns view vrf vpn101 user3`	Displays information about a particular DNS view, a group of views (with the same view name or associated with the same VRF), or all configured DNS views.
Step 3	**show ip dns server** [**vrf** *vrf-name*] [**default** | *view-name*] **Example:** `Device# show ip dns server vrf vpn101 user3`	Displays information from name server cache.
Step 4	**clear ip dns servers**	Cleans up server from name server cache.

Defining a DNS View List

Perform this task to define an ordered list of DNS views with optional, additional usage restrictions for each view list member. The device uses a DNS view list to select the DNS view that will be used to handle a DNS query.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns view-list** *view-list-name*
4. **ip dns name-list** [*number*] [*permit/deny*] [*name*]
5. **view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
6. **restrict name-group** *name-list-number*
7. **restrict source access-group** *acl-number*
8. **exit**
9. **end**
10. **show ip dns view-list** *view-list-name*
11. **show ip dns name-list** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip dns view-list** *view-list-name* **Example:** `Device(config)# ip dns view-list userlist5`	Defines a DNS view list and enters DNS view list configuration mode.
Step 4	**ip dns name-list** [*number*] [*permit/deny*] [*name*] **Example:** `Device(config)# ip dns name-list 10`	Defines a DNS name list and enters DNS name list configuration mode.
Step 5	**view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# view vrf vpn101 user5 10`	Defines a DNS view list member and enters DNS view list member configuration mode.
Step 6	**restrict name-group** *name-list-number* **Example:** `Device(cfg-dns-view-list-member)# restrict name-group 500`	(Optional) Specifies that this DNS view list member cannot be used to respond to a DNS query unless the query hostname matches a permit clause in the specified DNS name list and none of the deny clauses. • To define a DNS name list entry, use the **ip dns name-list** command.
Step 7	**restrict source access-group** *acl-number* **Example:** `Device(cfg-dns-view-list-member)# restrict access-group 99`	(Optional) Specifies that this DNS view list member cannot be used to respond to a DNS query unless the source IP address of the DNS query matches the specified standard ACL. • To define a standard ACL entry, use the **access-list** command.
Step 8	**exit** **Example:** `Device(cfg-dns-view-list-member)# exit`	Exits DNS view list member configuration mode. • To add another view list member to the list, go to Step 4.
Step 9	**end** **Example:** `Device(cfg-dns-view-list)# end`	Returns to privileged EXEC mode.
Step 10	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.

	Command or Action	Purpose
Step 11	**show ip dns name-list** *number* **Example:** `Device# show ip dns name-list 5`	Displays information about a particular DNS name list or all configured DNS name lists.

Modifying a DNS View List

To provide for efficient management of the order of the members in a view list, each view list member definition includes the specification of the position of that member within the list. That is, the order of the members within a view list is defined by explicit specification of position values rather than by the order in which the individual members are added to the list. This enables you to perform either of the following tasks without having to remove all the view list members and then redefine the view list membership in the desired order:

Adding a Member to a DNS View List Already in Use

Perform this optional task if you need to add another member to a DNS view list that is already in use.

For example, suppose the DNS view list named userlist5 is already defined and in use as a default view list or as an interface-specific view list. Assume that the list consists of the following members:

- DNS view user1 with position number 10
- DNS view user2 with position number 20
- DNS view user3 with position number 30

If you need to add DNS view user4 as the second member of the list, add that view to the list with a position number value from 11 to 19. You do not need to remove the three existing members and then add all four members to the list in the desired order.

SUMMARY STEPS

1. **enable**
2. **show ip dns view-list** *view-list-name*
3. **configure terminal**
4. **ip dns view-list** *view-list-name*
5. **view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
6. **end**
7. **show ip dns view-list** *view-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip dns view-list** *view-list-name* **Example:** `Device(config)# ip dns view-list userlist5`	Defines a DNS view list and enters DNS view list configuration mode.
Step 5	**view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# view user4 15`	Defines a DNS view list member and enters DNS view list member configuration mode.
Step 6	**end** **Example:** `Device(cfg-dns-view-list-member)# end`	Returns to privileged EXEC mode.
Step 7	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.

Changing the Order of the Members of a DNS View List Already in Use

Perform this optional task if you need to change the order of the members of a DNS view list that is already in use.

For example, suppose the DNS view list named userlist5 is already defined and in use as a default view list or as an interface-specific view list. Assume that the list consists of the following members:

- DNS view user1 with position number 10
- DNS view user2 with position number 20
- DNS view user3 with position number 30

If you want to move DNS view user1 to the end of the list, remove that view from the list and then add it back to the list with a position number value greater than 30. You do not need to remove the three existing members and then add the members back to the list in the desired order.

SUMMARY STEPS

1. **enable**
2. **show ip dns view-list** *view-list-name*
3. **configure terminal**
4. **ip dns view-list** *view-list-name*
5. **no view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
6. **view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number*
7. **end**
8. **show ip dns view-list** *view-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip dns view-list** *view-list-name* **Example:** `Device(config)# ip dns view-list userlist5`	Defines a DNS view list and enters DNS view list configuration mode.
Step 5	**no view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# no view user1 10`	Removes a DNS view list member from the list.
Step 6	**view** [**vrf** *vrf-name*] {**default** | *view-name*} *order-number* **Example:** `Device(cfg-dns-view-list)# view user1 40`	Defines a DNS view list member and enters DNS view list member configuration mode.
Step 7	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(cfg-dns-view-list-member)# end`	
Step 8	**show ip dns view-list** *view-list-name* **Example:** `Device# show ip dns view-list userlist5`	Displays information about a particular DNS view list or all configured DNS view lists.

Specifying the Default DNS View List for the DNS Server of the Device

Perform this task to specify the default DNS view list for the device's DNS server. The device uses the default DNS view list to select a DNS view to use to handle an incoming DNS query that arrives on an interface for which no interface-specific DNS view list has been defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns server view-group** *name-list-number*
4. **exit**
5. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dns server view-group** *name-list-number* **Example:** `Device(config)# ip dns server view-group 500`	Configures the default DNS view list for the device's DNS server.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 5	**show running-config** **Example:** `Device# show running-config`	Displays information about how DNS view lists are applied. The default DNS view list, if configured, is listed in the default DNS view information as the argument for the **ip dns server view-group** command.

Specifying a DNS View List for a Device Interface

Perform this optional task if you need to specify a DNS view list for a particular device interface. The device uses that view list to select a DNS view to use to handle a DNS query that arrives on that interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface*
4. **ip dns view-group** *view-list-name*
5. **end**
6. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface* **Example:** `Device(config)# interface ATM2/0`	Configures an interface type and enter interface configuration mode so that the specific interface can be configured.
Step 4	**ip dns view-group** *view-list-name* **Example:** `Device(config-if)# ip dns view-group userlist5`	Configures the DNS view list for this interface on the device.
Step 5	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 6	**show running-config** **Example:** `Device# show running-config`	Displays information about how DNS view lists are applied. Any DNS view lists attached to interfaces are listed in the information for each individual interface, as the argument for the **ip dns view-group** command.

Specifying a Source Interface to Forward DNS Queries

Perform this optional task if you need to specify a source interface to forward the DNS queries.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*}
4. **domain resolver source-interface** *interface-type number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dns view** [**vrf** *vrf-name*] {**default** | *view-name*} **Example:** `Device(config)# ip dns view vrf vpn32 user3`	Creates the DNS view of the specified name associated with the specified VRF instance and then enters DNS view configuration mode.
Step 4	**domain resolver source-interface** *interface-type number* **Example:** `Device(cfg-dns-view)# domain resolver source-interface fastethernet 0/0`	Sets the source IP address of the DNS queries for the DNS resolver functionality.
Step 5	**end** **Example:**	(Optional) Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	

Configuration Examples for DNS

Example: Creating a Domain List with Alternate Domain Names

The following example establishes a domain list with several alternate domain names:

```
ip domain list csi.com
ip domain list telecomprog.edu
ip domain list merit.edu
```

Example: Mapping Host Names to IP Addresses

The following example configures the host-name-to-address mapping process. IP DNS-based translation is specified, the addresses of the name servers are specified, and the default domain name is given.

```
! IP DNS-based host name-to-address translation is enabled
ip domain lookup
! Specifies hosts 192.168.1.111 and 192.168.1.2 as name servers
ip name-server 192.168.1.111 192.168.1.2
! Defines cisco.com as the default domain name the device uses to complete
! Set the name for unqualified host names
ip domain name cisco.com
```

Example: Customizing DNS

The following example shows the ip dns servers.

```
show ip dns server

   IP             VRF      TTL(s)   RTT(ms)  RTO(ms)  EDNS  DNSSEC  RECURSION
-----------------------------------------------------------------------------
2::1              red      628      1451     1451      Yes     Yes     Yes
172.168.10.1               875      1787     1787      Yes     Yes     Yes
2.2.2.1           red      606      1447     1447      Yes     Yes     Yes
1::1                       207      300      300       Yes     Yes     Yes
1.1.1.1                    179      242      242       Yes     Yes     Yes
```

Example: Split DNS View Lists Configured with Different View-use Restrictions

The following example shows how to define two DNS view lists, userlist1 and userlist2. Both view lists comprise the same three DNS views:

- DNS view user1 that is associated with the usergroup10 VRF
- DNS view user2 that is associated with the usergroup20 VRF
- DNS view user3 that is associated with the usergroup30 VRF

Both view lists contain the same DNS views, specified in the same order:

```
ip dns view-list userlist15
 view vrf usergroup100 user1 10
  restrict name-group 121
  exit
view vrf usergroup200 user2 20
  restrict name-group 122
  exit
view vrf usergroup300 user3 30
  restrict name-group 123
  exit
!
exit
ip dns view-list userlist16
 view vrf usergroup100 user1 10
  restrict name-group 121
  restrict source access-group 71
  exit
 view vrf usergroup200 user2 20
  restrict name-group 122
  restrict source access-group 72
  exit
 view vrf usergroup300 user3 30
  restrict name-group 123
  restrict source access-group 73
  exit
exit
```

The two DNS view lists differ, though, in the usage restrictions placed on their respective view list members. DNS view list userlist15 places only query hostname restrictions on its members while view list userlist16 restricts each of its members on the basis of the query hostname and the query source IP address:

- Because the members of userlist15 are restricted only based on the VRF from which the query originates, userlist15 is typical of a view list that can be used to select a DNS view for handling DNS requests from internal clients.
- Because the members of userlist16 are restricted not only by the query VRF and query hostname but also by the query source IP address, userlist16 is typical of a view list that can be used to select a DNS view for handling DNS requests from external clients.

Additional References for Configuring DNS

Related Documents

Related Topic	Document Title
Master Command List	Cisco IOS Master Command List
IP Addressing Services Command Reference	Cisco IOS IP Addressing Services Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1348	DNS NSAP Resource Records

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Configuring DNS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Configuring DNS

Feature Name	Releases	Feature Configuration Information
Configuring DNS	Cisco IOS XE Release 2.1	The Domain Name System (DNS) is a distributed database in which you can map host names to IP addresses through the DNS protocol from a DNS server. Each unique IP address can have an associated host name. The Cisco IOS XE software maintains a cache of host name-to-address mappings. This cache speeds the process of converting names to addresses.
	Cisco IOS XE Release 3.13S	The following commands were introduced or modified: **debug ip domain**, **debug ip domain replies**.
	Cisco IOS XE Release 3.16S	The following commands were introduced or modified: **dns trust**, **clear ip dns servers**.

CHAPTER 3

VRF-Aware DNS

The VRF-Aware DNS feature enables the configuration of a Virtual Private Network (VPN) routing and forwarding instance (VRF) table so that the domain name system (DNS) can forward queries to name servers using the VRF table rather than the named DNS server in the global IP address space. This feature allows DNS requests to be resolved within the appropriate Multiprotocol Label Switching (MPLS) VPN.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About VRF-Aware DNS

Domain Name System

Domain Name System (DNS) is a standard that defines a domain naming procedure used in TCP/IP. A domain is a hierarchical separation of the network into groups and subgroups with domain names identifying the structure. The named groups consist of named objects, usually devices like IP hosts, and the subgroups are domains. DNS has three basic functions:

- Name space: This function is a hierarchical space organized from a single root into domains. Each domain can contain device names or more specific information. A special syntax defines valid names and identifies the domain names.

- Name registration: This function is used to enter names into the DNS database. Policies are outlined to resolve conflicts and other issues.
- Name resolution: This function is a distributed client and server name resolution standard. The name servers are software applications that run on a server and contain the resource records (RRs) that describe the names and addresses of those entities in the DNS name space. A name resolver is the interface between the client and the server. The name resolver requests information from the server about a name. A cache can be used by the name resolver to store learned names and addresses.

A DNS server can be a dedicated device or a software process running on a device. The server stores and manages data about domains and responds to requests for name conflict resolutions. In a large DNS implementation, there can be a distributed database over many devices. A server can be a dedicated cache.

VRF Mapping and VRF-Aware DNS

To keep track of domain names, IP has defined the concept of a name server, whose job is to hold a cache (or database) of names appended to IP addresses. The cached information is important because the requesting DNS will not need to query for that information again, which is why DNS works well. If a server had to query each time for the same address because it had not saved any data, the queried servers would be flooded and would crash.

A gateway for multiple enterprise customers can be secured by mapping the remote users to a VRF domain. Mapping means obtaining the IP address of the VRF domain for the remote users. By using VRF domain mapping, a remote user can be authenticated by a VRF domain-specific AAA server so that the remote-access traffic can be forwarded within the VRF domain to the servers on the corporate network.

To support traffic for multiple VRF domains, the DNS and the servers used to resolve conflicts must be VRF aware. VRF aware means that a DNS subsystem will query the VRF name cache first, then the VRF domain, and store the returned RRs in a specific VRF name cache. Users are able to configure separate DNS name servers per VRF.

VRF-aware DNS forwards queries to name servers using the VRF table. Because the same IP address can be associated with different DNS servers in different VRF domains, a separate list of name caches for each VRF is maintained. The DNS looks up the specific VRF name cache first, if a table has been specified, before sending a query to the VRF name server. All IP addresses obtained from a VRF-specific name cache are routed using the VRF table.

How to Configure VRF-Aware DNS

Defining a VRF Table and Assigning a Name Server to Enable VRF-Aware DNS

Perform this task to define a VRF table and assign a name server.

A VRF-specific name cache is dynamically created if one does not exist whenever a VRF-specific name server is configured by using the **ip name-server vrf**command option or a permanent name entry is configured by using the **ip host vrf** command option. The VRF name cache is removed whenever all name server and permanent entries in the VRF are disabled.

It is possible that multiple name servers are configured with the same VRF name. The system will send queries to those servers in turn until any of them responds, starting with the server that sent a response the last time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*]
7. **ip domain lookup** [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip vrf** *vrf-name* **Example:** `Router(config)# ip vrf vpn1`	Defines a VRF table and enters VRF configuration mode. • The *vrf-name* argument can be up to 32 characters.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 100:21`	Creates routing and forwarding tables for a VRF.
Step 5	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode.
Step 6	**ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*] **Example:** `Router(config)# ip name-server vrf vpn1 172.16.1.111 172.16.1.2`	Assigns the address of one or more name servers to a VRF table to use for name and address resolution. • The **vrf** keyword is optional but must be specified if the name server is used with VRF. The *vrf-name* argument assigns a name to the VRF.
Step 7	**ip domain lookup** [**vrf** *vrf-name*] **Example:** `Router(config)# ip domain lookup vrf`	(Optional) Enables DNS-based address translation. • DNS is enabled by default. You only need to use this command if DNS has been disabled.

Mapping VRF-Specific Hostnames to IP Addresses

Perform this task to map VRF-specific hostnames to IP addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip domain name** [**vrf** *vrf-name*] *name*
   - **ip domain list** [**vrf** *vrf-name*] *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **ip domain name** [**vrf** *vrf-name*] *name*
	• **ip domain list** [**vrf** *vrf-name*] *name*
	Example:
	`Device(config)# ip domain name vrf vpn1 cisco.com`
	Example:
	`Device(config)# ip domain list vrf vpn1 cisco.com`
		Defines a default domain name that the software will use to complete unqualified hostnames.
		or
		Defines a list of default domain names to complete unqualified hostnames.
		• You can specify a default domain name that the software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up.
		• The **vrf** keyword and *vrf-name* argument specify a default VRF domain name.
		• The **ip domain list** command can be entered multiple times to specify more than one domain name to append when doing a DNS query. The system will append each in turn until it finds a match.

Configuring a Static Entry in a VRF-Specific Name Cache

Perform this task to configure a static entry in a VRF-specific name cache.

A VRF-specific name cache is dynamically created if one does not exist whenever a name server is configured for the VRF by using the **ip name-server vrf**command option or a permanent name entry is configured by using the **ip host vrf** command option. The VRF name cache is removed whenever all name server and permanent entries in the VRF are disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip host vrf** [*vrf-name*] *name*[*tcp-port*] *address1*[*address2 ... address8*] [*mx ns srv*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip host vrf** [*vrf-name*] *name*[*tcp-port*] *address1*[*address2 ... address8*] [*mx ns srv*]
	Example:
	`Device(config)# ip host vrf vpn3 company1.com 172.16.2.1`
	`Device(config)# ip host test mx 1 mx_record`
	`Device(config)# ip host test ns ns_record`
	`Device(config)# ip host test srv 0 0 0 srv_record`
		Defines a static hostname-to-address mapping in the host cache.
		• The IP address of the host can be an IPv4 or IPv6 address, and the IP address can be associated with a Virtual Private Network (VPN) routing and forwarding (VRF) instance.
		• If the **vrf** keyword and *vrf-name* arguments are specified, then a permanent entry is created only in the VRF-specific name cache.
		• Mail exchanger (mx) identifies the mail server that is responsible for handling e-mails for a given domain name.
		• Name server (ns) state the authoritative name servers for the given domain.
		• Service (srv) records specifies the location of a service.

Verifying the Name Cache Entries in the VRF Table

Perform this task to verify the name cache entries in the VRF table.

SUMMARY STEPS

1. **enable**
2. **show hosts** [**vrf** *vrf-name*] {**all**| *hostname*} [**summary**]
3. **clear host** [**vrf** *vrf-name*] {**all**| *hostname*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show hosts** [**vrf** *vrf-name*] {**all**| *hostname*} [**summary**] **Example:** `Device# show hosts vrf vpn2`	• Displays the default domain name, the style of name lookup service, a list of name server hosts, the cached list of hostnames and addresses, and the cached list of hostnames and addresses specific to a particular Virtual Private Network (VPN). • The **vrf** keyword and *vrf-name* argument only display the entries if a VRF name has been configured. • If you enter the **show hosts** command without specifying any VRF, only the entries in the global name cache will display.
Step 3	**clear host** [**vrf** *vrf-name*] {**all**| *hostname*} **Example:** `Device# clear host vrf vpn2`	(Optional) Deletes entries from the hostname-to-address global address cache or VRF name cache.

Configuration Examples for VRF-Aware DNS

Example: VRF-Specific Name Server Configuration

The following example shows how to specify a VPN named vpn1 with the IP addresses of 172.16.1.111 and 172.16.1.2 as the name servers:

```
ip name-server vrf vpn1 172.16.1.111 172.16.1.2
```

Example: VRF-Specific Domain Name List Configuration

The following example shows how to add several domain names to a list in vpn1 and vpn2. The domain name is only used for name queries in the specified VRF.

```
ip domain list vrf vpn1 company.com
ip domain list vrf vpn2 school.edu
```

If there is no domain list, the domain name that you specified with the **ip domain name** global configuration command is used. If there is a domain list, the default domain name is not used. The **ip domain list** command is similar to the **ip domain name** command, except that with the **ip domain list**command you can define a list of domains, each to be tried in turn until a match is found.

VRF-Specific Domain Name Configuration Example

The following example shows how to define cisco.com as the default domain name for a VPN named vpn1. The domain name is only used for name queries in the specified VRF.

```
ip domain name vrf vpn1 cisco.com
```

Any IP hostname that does not contain a domain name (that is, any name without a dot) will have the dot and cisco.com appended to it before being looked up.

VRF-Specific IP Host Configuration Example

The following example shows how to define two static hostname-to-address mappings in the host cache for vpn2 and vpn3:

```
ip host vrf vpn2 host2 10.168.7.18
ip host vrf vpn3 host3 10.12.0.2
```

Additional References

Related Documents

Related Topic	Document Title
VRF-aware DNS configuration tasks: Enabling VRF-aware DNS, mapping VRF-specific hostnames to IP addresses, configuring a static entry in a VRF-specific hostname cache, and verifying the hostname cache entries in the VRF table	"VRF-Aware DNS" module
DNS configuration tasks	"Configuring DNS" module
DNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Addressing Services Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VRF-Aware DNS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for DNS

Feature Name	Releases	Feature Configuration Information
VRF-Aware DNS	Cisco IOS XE Release 2.1	The VRF-Aware DNS feature enables the configuration of a Virtual Private Network (VPN) routing and forwarding instance (VRF) table so that the domain name system (DNS) can forward queries to name servers using the VRF table rather than the named DNS server in the global IP address space. This feature allows DNS requests to be resolved within the appropriate Multiprotocol Label Switching (MPLS) VPN.

CHAPTER 4

Local Area Service Discovery Gateway

The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across Layer 3 (L3) boundaries. An mDNS gateway will be able to provide transport for service discovery across L3 boundaries by filtering, caching and extending services from one subnet to another. Prior to implementation of this feature, mDNS was limited in scope to within a subnet due to the use of link-local scoped multicast addresses. This feature enhances Bring Your Own Device (BYOD).

Caution Extension of services should be done with proper care. Generally, only specific services should be extended. Service names should be unique in the network to avoid duplicate name conflicts.

See Feature Information for Service Discovery Gateway section to check feature availability for your platform release version.

Information About Service Discovery Gateway

Service Announcement Redistribution and Service Extension

Redistribution of announcements is the actual forwarding of announcements and query responses while service extension is the capability of proxying services between subnets. The actual replication of the service announcement can help to speed up the visibility of newly announced services and also a service's withdrawal if a service or device is turned off.

Note Extension of services such as printers or Apple TV works fine without actual replication of service announcements. The Service Discovery Gateway will cache announcements, queries and their responses in the cache. If another device queries for a service, the Service Discovery Gateway will be able to provide an answer from its cache.

Enable the **redistribution mdns-sd** command only on a per-interface basis, and only if it is actually required. You must ensure that there are no loops in the network topology corresponding to the interface for which service announcement redistribution is being enabled. A loop can lead to a broadcast storm.

Redistribution of service announcement information cannot be done globally. You can enable redistribution of service information only at the interface level.

Extending Services Across Subnets—An Overview

You need to enable a multicast Domain Name System (mDNS) gateway to extend services across subnet boundaries. You can enable an mDNS gateway for a device or for an interface. You must enable routing of services for the device before enabling it at the interface level. After the mDNS gateway is enabled on a device or interface, you can extend services across subnet boundaries.

To extend services across subnets, you must do the following:

1. Set Filter Options to Extend Services Across Subnets—You can allow services such as printer services to be accessed across subnets. If printer x is available on interface 1, users on interface 2 can use printer x without configuring the printer on their local systems.
2. Extend Services Across Subnets—The filter created in Step 1 should be applied on the interfaces 1 and 2. Only then can users on other interfaces access the printer service.

For the sample scenario where a printer service is accessible by clients on other interfaces, you must apply these filters:

- On the interface where the printer service is available (IN filter) —You want to allow the printer service *into* the mDNS cache, so that it can be accessed by users on other subnets.
- On the interface where the printer service is available (OUT filter)—Since clients on other interfaces will access the service (printer x, for example), you should allow queries coming from the device (OUT filter, from the device's point of view).
- On each interface where clients reside (IN filter)—For clients on other interfaces (subnets) wanting to access the printer service, you must allow queries from users into the mDNS cache (IN filter).

Remember Applying the IN filter means that you are allowing the printer service into the device mDNS cache, and other interfaces can access it. Applying the OUT filter means that you are allowing the queries out of the cache so that queries from clients on other interfaces can reach the printer interface. On other client-facing interfaces, the IN filter is applied to allow queries in.

Note

- Filters can be applied at the global level and at the interface level. Filters applied at the interface level takes precedence over the filters applied at the global level.
- The term 'service discovery information' refers to services (printer services, etc), queries (queries for printer services, etc, from one interface to the other), announcements (printer service is removed, etc), and service-instances (a specific service—printer x, Apple TV 3, etc) that you want to extend across subnets.

Set Filter Options to Extend Services Across Subnets

You can set filter options to allow services such as printer services into or out of a device or interface. You can also permit or prohibit queries, announcements, services learnt from an interface, specific service–instances, and locations. Use the **service-list mdns-sd** command to create a service-list and set filter options.

You need to create a service-list and use filter options within it. While creating a service-list, use one of the following options:

- The **permit** option permits specific services, announcements and service-instances across subnets.
- The **deny** option restricts services, announcements and service-instances from being transported across subnets.
- The **query** option is provided to browse services. For example, if you want to browse printer services periodically, then you can create a service-list with the **query** option, and add the printer service to the query. When you set a period for the query, the service entries are refreshed in the cache memory.

You must mention a sequence number when using the **permit** or **deny** option. The filtering is done sequentially, in the ascending order. The same service-list can be associated with multiple sequence numbers. Within a sequence, match statements (commands) must be used to specify what needs to be filtered. Generally, match statements are used to filter queries (for example, queries from clients to find printer and fax services), announcements (new service is added, and so on), specific service–instances, types of service such as printer services (so that the service is allowed into the cache for use), services available for a specific interface (printers and Apple TVs associated with a VLAN), and locations.

Note A service-list by itself does not contain any services. You must specify a service type in the match statement when setting filter options to allow or prohibit services. (For example, '_ipp._tcp' is the service type for an IPP printing service running over TCP).

Sample scenario - Consider a device is in a client segment. The goal is to allow the following on the device:

- All queries from clients to the device.
- Printer services to clients on other subnets.

The following example explains how to achieve the goal:

```
!
service-list mdns-sd mixed permit 10
  match message-type query
!
service-list mdns-sd mixed permit 20
  match message-type announcement
  match service-type _ipps._tcp.local
!
```

In the above example, a service-list called 'mixed' is created and the **permit** option is used twice—to filter queries and to filter printer services and announcements. The filtering is done in the sequence given below:

- Sequence 10 - A match statement is used to filter queries.
- Sequence 20 - Match statements are used to filter announcements and printer services.

The match statement in Sequence 10 sets a filter for queries on the device, but does not specify that queries be allowed *into* the device. To allow queries from clients, the filter needs to be applied on the interface in the IN direction. The example is displayed in the Extend Services Across Subnets section.

Similarly, the match statements in Sequence 20 sets a filter for announcements and printer services on the device, but does not specify that they be allowed *into* the device. To allow announcements and printer services into the device, the filter needs to be applied on the required interfaces in the IN direction. The example is displayed in the Extend Services Across Subnets section.

If neither the **permit** option nor the **deny** option is used, the default action is to disallow services from being transported to other subnets.

Browsing services periodically—Service-lists of the type **query** can be used to browse services. Such queries are called active queries. Active queries periodically send out requests for the services specified within the query on all interfaces. As services have a specific Time to Live (TTL) duration, active queries can help to keep services fresh in the cache memory.

In the following example, a service-list named 'active-query' is created and the service-list is of the type **query**. Services such as printer services are specified within the query, and these are the services that we want to extend. Typically, these services would match the services that have been configured as 'permitted' services in the IN filter.

```
!
service-list mdns-sd active-query query
  service-type _universal._sub._ipp._tcp
  service-type _ipp._tcp.local
  service-type _ipps._tcp.local
  service-type _raop._tcp.local
!
```

The purpose of an active query and a query associated with a match statement is different. When you enable an active query, services are browsed periodically. A query is used in a match statement to permit or prohibit queries (not active queries) on the interface.

Note

- Service-list creation can only be used globally and cannot be used at the interface level.
- You can create a new service-instance of a specific service-type using the **service-instance mdns-sd** command.
- A service end–point (such as a printer, fax, and so on) sends unsolicited announcements when a service starts up. After that, it sends unsolicited announcements whenever a network change event occurs (such as, an interface coming up or going down, and so on). The device always responds to queries.

Remember

Filtering only sets filter options and specifies that certain services need to be filtered. You must *apply* the filters on an interface for the services, queries, or announcements to actually be permitted or prohibited on the interface. To know about applying filters and the other available service discovery configuration options, refer the Extend Services Across Subnets section.

Extend Services Across Subnets

You must have set filter options for the device before extending services across subnets. If you have set filter options for specific services and other service discovery information to be allowed, prohibited or queried periodically, you can apply the filters for an interface.

Before applying filters, note the following:

- You must enable multicast Domain Name System (mDNS) on a device to apply filter options. You can enable mDNS using the command **service-routing mdns-sd**
- Since you might want to allow services into the device or prohibit services from being learnt on an interface, you must apply the filter in the needed direction. The options **IN** and **OUT** perform the desired actions on the interface.
- Typically, a service-policy is applied on an interface. Global service-policies are optional and affect all L3 interfaces.

Sample scenario - A device is in a client segment and the goal is to allow the following between the device interfaces:

- All queries from clients to the device.
- Printer services.

A note about filter options - Filter options have been set for the above scenario by creating a service-list called 'mixed' and adding filter options to it. (see Set Filter Options to Extend Services Across Subnets for more details). The following example explains how to apply the filters:

```
!
interface Ethernet0/0
 description *** (wireless) Clients here plus some printers
 ip address 172.16.33.7 255.255.255.0
 service-routing mdns-sd
  service-policy mixed IN
!
interface Ethernet0/3
 description *** (wireless) Clients here plus some printers
 ip address 172.16.57.1 255.255.255.0
 service-routing mdns-sd
  service-policy mixed IN
!
```

In the above example, service-routing is enabled on the interface and the filter options in the service-policy 'mixed' are applied in the **IN** direction. In other words, all queries and printer services will be allowed into the device, from the interfaces Ethernet 0/0 and Ethernet 0/3.

Sample scenario for browsing specific services - A service-list of the type **query** (called active query) has been created. It contains services that we want to browse periodically, such as printer services (see Set Filter Options to Extend Services Across Subnets for more details about creating an active query). To enable browsing of the services in the query, you must apply the active query for the device.

```
!
service-routing mdns-sd
 service-policy-query active-query 90
!
```

In the above example, the period is set to 90 seconds. The services within the active query are queried on all interfaces of the device after an interval of 90 seconds.

Note

- You can enable browsing of services for specific interfaces. If browsing of services is enabled globally, you can disable browsing of services on specific interfaces.
- Services are browsed specific to a device or interface by the mDNS process. So, the IN or OUT option is not relevant for browsing of services.

You can use the following options after enabling mDNS on a device or interface.

Purpose	Use this Command **Note** The complete syntax is provided in the corresponding task.	Global and Interface Configuration Options
For a service-list, apply a filter to allow or prohibit services.	**service-policy**	Global and interface levels.
Set some part of the system memory for cache.	**cache-memory-max**	Global level.
Configure an active query and the query period so that specified services are queried periodically.	**service-policy-query**	
Designate a specific device or interface in a domain for routing mDNS announcement and query information.	**designated-gateway**	Global and interface levels.
Access services in the proximity of the device. **Note** Service policy proximity filtering functionality is only available on wireless devices and their interfaces.	**service-policy-proximity**	Global and interface levels.
Configure service-type enumeration period for the device.	**service-type-enumeration period**	Global level.
Specify an alternate source interface for outgoing mDNS packets on a device.	**source-interface**	Global level.
Configure the maximum rate limit of incoming mDNS packets for a device.	**rate-limit**	Global level.

Speed up visibility of newly announced services and withdrawal of services when a service or device is turned off.	**redistribute**	Interface level.

How to Configure Service Discovery Gateway

Setting Filter Options for Service Discovery

Before you begin

Ensure that you permit a query or announcement when you set filter options. If you do not use a **permit** option and only use **deny** options, you will not be able to apply the filter.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**}
4. **match message-type** {**announcement** | **any** | **query**}
5. **match service-instance** {*instance-name* | **any** | **query**}
6. **match service-type** *mDNS-service-type-string*
7. **match location civic** *civic-location-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**} **Example:** `Device(config)# service-list mdns-sd sl1 permit 3`	Enters mdns service discovery service-list mode. • Creates a service-list and applies a filter on the service-list according to the **permit** or **deny** option applied to the sequence number. Or

	Command or Action	Purpose
	Or `Device(config)# service-list mdns-sd sl4 query`	• Creates a service-list and associates a query for the service-list name if the **query** option is used. **Remember** When you set filter options, ensure that you permit a query or announcement for a service-list. If you do not use a **permit** option and only use **deny** options, you will not be able to apply the filter.
Step 4	**match message-type** {**announcement** | **any** | **query**} **Example:** `Device(config-mdns-sd-sl)# match message-type announcement`	Configures parameters for a service-list based on a service announcement or query. **Note** You cannot use the **match** command if you have used the **query** option. The **match** command can be used only for the **permit** or **deny** option.
Step 5	**match service-instance** {*instance-name* | **any** | **query**} **Example:** `Device(config-mdns-sd-sl)# match service-instance printer-3`	Configures parameters for a service-list based on a service-instance or query.
Step 6	**match service-type** *mDNS-service-type-string* **Example:** `Device(config-mdns-sd-sl)# match service-type _ipp._tcp.local`	Configures parameters for a service-list based on a service-type.
Step 7	**match location civic** *civic-location-name* **Example:** `Device(config-mdns-sd-sl)# match location civic location3`	Configures parameters for a service-list based on a civic location.
Step 8	**exit** **Example:** `Device(config-mdns-sd-sl)# exit`	Exits mdns service discovery service-list mode, and returns to global configuration mode.

What to do next

Apply filters on an interface for the services, queries, or announcements to actually be permitted or prohibited on the interface.

Applying Service Discovery Filters and Configuring Service Discovery Parameters

After enabling multicast Domain Name System (mDNS) gateway for a device, you can apply filters (IN-bound filtering or OUT-bound filtering) and active queries by using **service-policy** and **service-policy-query** commands, respectively.

Note Steps 5 to 11 are mDNS Service Discovery configuration options. The steps are optional and not meant to be used in any specific order.

Before you begin

You must set filter options for the device before applying filters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-routing mdns-sd**
4. **service-policy** *service-policy-name* {**IN** | **OUT**}
5. **cache-memory-max** *cache-config-percentage*
6. **service-policy-query** *service-list-name query-period*
7. **designated-gateway enable** [**ttl** *duration*]
8. **service-policy-proximity** *service-list-name* [**limit** *number-of-services*]
9. **service-type-enumeration period** *period-value*
10. **source-interface** *type number*
11. **rate-limit in** *maximum-rate-limit*
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-routing mdns-sd** **Example:**	Enables mDNS gateway functionality for a device and enters multicast DNS configuration (config-mdns) mode.

	Command or Action	Purpose
	`Device(config)# service-routing mdns-sd`	
Step 4	**service-policy** *service-policy-name* {**IN** | **OUT**} **Example:** `Device(config-mdns)# service-policy sl1 IN`	For a service-list, applies a filter on incoming service discovery information (IN-bound filtering) or outgoing service discovery information (OUT-bound filtering). **Note** Global service-policies are optional and effect all L3 interfaces. Typically, a service-policy is applied on an interface.
Step 5	**cache-memory-max** *cache-config-percentage* **Example:** `Device(config-mdns)# cache-memory-max 20`	Sets some part of the system memory (in percentage) for cache. **Note** By default, 10% of the system memory is set aside for cache. You can override the default value by using this command.
Step 6	**service-policy-query** *service-list-name* *query-period* **Example:** `Device(config-mdns)# service-policy-query sl4 100`	Creates an active query and configures the service-list-query period.
Step 7	**designated-gateway** **enable** [**ttl** *duration*] **Example:** `Device(config-mdns)# designated-gateway enable`	Designates the device to route mDNS announcement and query information for the domain.
Step 8	**service-policy-proximity** *service-list-name* [**limit** *number-of-services*] **Example:** `Device(config-mdns)# service-policy-proximity sl1 limit 10`	Configures service policy proximity filtering on the device. • Service policy proximity filtering is only available for wireless clients and is based on Radio Resource Management (RRM). Wired clients and services are not affected by the limit. • The default value for the maximum number of services that can be returned is 50.
Step 9	**service-type-enumeration period** *period-value* **Example:** `Device(config-mdns)# service-type-enumeration period 45`	Configures service-type enumeration period for the device.
Step 10	**source-interface** *type number* **Example:**	Specifies an alternate source interface for outgoing mDNS packets on a device.
Step 11	**rate-limit** **in** *maximum-rate-limit* **Example:** `Device(config-mdns)# rate-limit in 80`	Configures the maximum rate limit of incoming mDNS packets for a device.

	Command or Action	Purpose
Step 12	**exit** **Example:** `Device(config-mdns)# exit`	Exits multicast DNS configuration mode, and returns to global configuration mode.

Applying Service Discovery Filters for an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-routing mdns-sd**
5. **service-policy** *service-policy-name* {**IN** | **OUT**}
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters Interface multicast DNS configuration mode, and enables interface configuration.
Step 4	**service-routing mdns-sd** **Example:** `Device(config-if)# service-routing mdns-sd`	Enables mDNS gateway functionality for an interface and enters multicast DNS configuration (config-mdns) mode.
Step 5	**service-policy** *service-policy-name* {**IN** | **OUT**} **Example:** `Device(config-if-mdns-sd)# service-policy sl1 IN`	For a service-list, applies a filter on incoming service discovery information (IN-bound filtering) or outgoing service discovery information (OUT-bound filtering).

	Command or Action	Purpose
		Remember When you set filter options, ensure that you permit a query or announcement for a service-list. If you have not permitted a service, query, or announcement while setting filter options, then you will see this warning when you apply the filter: **Warning**: Please enable explicit service-list rule with the permit action to allow queries and responses.
Step 6	**exit** **Example:** `Device(config-if-mdns-sd)# exit`	Exits Interface multicast DNS configuration mode, and returns to interface configuration mode.

Creating a Service Instance

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **service-instance mdns-sd service** *instance-name* **regtype** *service-type* **domain** *name*
4. {**ipv4addr** | **ipv6addr**} *IP-address*
5. **port** *number*
6. **target-hostname** *host-name*
7. **txt** *text-record-name*
8. **priority** *value*
9. **weight** *value*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**service-instance mdns-sd service** *instance-name* **regtype** *service-type* **domain** *name* **Example:** `Device(config)# service-instance mdns-sd service printer-3 regtype _ipp._tcp.local domain tcp4`	Creates a service-instance of a specific service type and enters multicast Domain Name System (mDNS) service discovery service-instance (config-mdns-sd-si) mode. **Note** In this mode, you can configure various parameters for the service-instance. The subsequent steps show how to configure service-instance parameters.
Step 4	{**ipv4addr** | **ipv6addr**} *IP-address* **Example:** `Device(config-mdns-sd-si)# ipv4addr 209.165.200.230 255.255.255.0`	Specifies the IPv4 or IPv6 address of the port on which the service is available.
Step 5	**port** *number* **Example:** `Device(config-mdns-sd-si)# port 9100`	Specifies the port on which the service is available.
Step 6	**target-hostname** *host-name* **Example:** `Device(config-mdns-sd-si)# target-hostname fqdn-of-printer.example.com.`	Specifies the fully qualified domain name (FQDN) of the target host.
Step 7	**txt** *text-record-name* **Example:** `Device(config-mdns-sd-si)# txt _ipp._tcp.local=printer3`	Specifies the text record associated with the service instance. **Note** A TXT record is a type of DNS record that provides text information to sources outside your domain. Specify the text record in the format 'service-type=service-name'. To specify multiple records, use a semicolon (;) as a separator.
Step 8	**priority** *value* **Example:** `Device(config-mdns-sd-si)# priority 3`	(Optional) Specifies the priority value for the service-instance. The default priority value is zero.
Step 9	**weight** *value* **Example:** `Device(config-mdns-sd-si)# weight 20`	(Optional) Specifies the weight value for the service-instance. The default weight value is zero.

	Command or Action	Purpose
Step 10	**exit** **Example:** `Device(config-mdns-sd-si)# exit`	Exits multicast Domain Name System (mDNS) service discovery service-instance (config-mdns-sd-si) mode and enters global configuration mode.

Verifying and troubleshooting Service Discovery Gateway

Note The show and debug commands mentioned below are not in any specific order.

SUMMARY STEPS

1. **show mdns requests** [**detail** | [**type** *record-type*] [**name** *record-name*]]
2. **show mdns cache** [**interface** *type number* [**detail**] | [**name** *record-name*] [**type** *record-type*] [**detail**]]
3. **show mdns statistics** {**all** | **interface** *type number* | **service-list** *list-name* | [**cache** | **service-policy**] {**all** | **interface** *type number*} | **services orderby providers**}
4. **show mdns service-types** [**all** | **interface** *type number*]
5. **debug mdns** {**all** | **error** | **event** | **packet** | **verbose**}

DETAILED STEPS

Step 1 **show mdns requests** [**detail** | [**type** *record-type*] [**name** *record-name*]]

Example:

```
Device# show mdns requests detail

MDNS Outstanding Requests
======================================================
Request name  :   _ipp._tcp.local
Request type  :   PTR
Request class :   IN
```

This command displays information for outstanding multicast Domain Name System (mDNS) requests, including record name and record type information.

Step 2 **show mdns cache** [**interface** *type number* [**detail**] | [**name** *record-name*] [**type** *record-type*] [**detail**]]

Example:

Note You can use the **detail** keyword for a specific interface, record or type. You cannot use it independently with the **show mdns cache** command.

```
Device# show mdns cache

mDNS CACHE
=================================================================================================
[<NAME>]                                              [<TYPE>][<CLASS>] [<TTL>/Remaining] [Accessed]
```

```
 [If-index] [<RR Record Data>]

_services._dns-sd._udp.local                          PTR     IN      4500/4496             0
     3       _ipp._tcp.local

_ipp._tcp.local                                       PTR     IN      4500/4496             1
     3       printer1._ipp._tcp.local

printer1._ipp._tcp.local                              SRV     IN      120/116        1      3
     0       0       5678    much-WS.local

printer1._ipp._tcp.local                              TXT     IN      4500/4496             1
     3       (1)''

music-WS.local                                     A       IN      120/116        1      3
  192.168.183.1
```

This command displays mDNS cache information.

Step 3 **show mdns statistics** {**all** | **interface** *type number* | **service-list** *list-name* | [**cache** | **service-policy**] {**all** | **interface** *type number*} | **services orderby providers**}

Example:

```
Device# show mdns statistics all

mDNS Statistics
 mDNS packets sent       : 0
 mDNS packets received   : 31
 mDNS packets dropped    : 8
 mDNS cache memory in use: 64264(bytes)
```

This command displays mDNS statistics.

Step 4 **show mdns service-types** [**all** | **interface** *type number*]

Example:

```
Device# show mdns service-types

mDNS SERVICES
=================================
[<NAME>]          [<TTL>/Remaining] [If-name]
_ipp._tcp.local     4500/4496
```

This command displays mDNS statistics.

Step 5 **debug mdns** {**all** | **error** | **event** | **packet** | **verbose**}

Example:

```
Device# debug mdns all
```

This command enables all mDNS debugging flows.

Configuration Examples for Service Discovery Gateway

Example: Setting Filter Options for Service Discovery

The following example shows creation of a service-list sl1. The permit option is being applied on sequence number 3 and all services with message-type announcement are filtered and available for transport across various subnets associated with the device.

```
Device> enable
Device# configure terminal
Device(config)# service-list mdns-sd sl1 permit 3
Device(config-mdns-sd-sl)# match message-type announcement
Device(config-mdns-sd-sl)# exit
```

Example: Applying Service Discovery Filters and Configuring Service Discovery Parameters

```
Device> enable
Device# configure terminal
Device(config)# service-routing mdns-sd
Device(config-mdns)# service-policy serv-pol1 IN
Device(config-mdns)# cache-memory-max 20
Device(config-mdns)# service-policy-query sl-query1 100
Device(config-mdns)# designated-gateway enable
Device(config-mdns)# rate-limit in 80
Device(config-mdns)# exit
```

Example: Applying Service Discovery Filters for an Interface

Example: Setting Multiple Service Discovery Filter Options

The following example shows creation of filters using service-lists mixed, permit-most, permit-all, and deny-all. Then, the filters are applied at various interfaces, as required.

```
!
service-list mdns-sd mixed permit 10
 match message-type query
!
service-list mdns-sd mixed permit 20
 match message-type announcement
 match service-type _ipps._tcp.local
!
service-list mdns-sd mixed permit 30
 match message-type announcement
 match service-type _ipp._tcp.local
```

```
 match service-type _universal._sub._ipp._tcp
!
service-list mdns-sd mixed permit 40
 match message-type announcement
!
service-list mdns-sd mixed deny 50
!
service-list mdns-sd permit-most deny 10
 match service-type _sleep-proxy._udp.local
!
service-list mdns-sd permit-most permit 20
!
service-list mdns-sd permit-all permit 10
!
service-list mdns-sd deny-all permit 10
 match message-type query
!
service-list mdns-sd deny-all deny 20
!
service-list mdns-sd active-query query
 service-type _universal._sub._ipp._tcp.local
 service-type _ipp._tcp.local
 service-type _ipps._tcp.local
 service-type _raop._tcp.local
!
service-routing mdns-sd
 service-policy-query active-query 900
!
!
interface Ethernet0/0
 description *** (wireless) Clients here plus some printers or aTVs
 ip address 172.16.33.7 255.255.255.0
 service-routing mdns-sd
  service-policy mixed IN
  service-policy permit-all OUT
!
interface Ethernet0/1
 description *** AppleTVs, Print Servers here
 ip address 172.16.57.1 255.255.255.0
 service-routing mdns-sd
  service-policy permit-most IN
  service-policy permit-all OUT
!
interface Ethernet0/2
 description *** Clients only, we don't want to learn anything here
 ip address 172.16.58.1 255.255.255.0
 service-routing mdns-sd
  service-policy deny-all IN
  service-policy permit-all OUT
!
interface Ethernet0/3
 no ip address
 shutdown
!
```

In the above example, the service-lists are:

- permit-all - As the name suggests, this service-list permits all resource records, and should be used with care. This is typically applied in the OUT direction; allows the cache to respond to all requests regardless of query content or query type.
- permit-most - This allows anything in, except for sleep-proxy services. This is because extending sleep-proxy services causes an issue with devices that register with a sleep proxy across the

Service Discovery Gateway. Due to split horizon, the real (sleeping) device won't be able to re-register its services when waking up again when its pointer (PTR) record is pointing to the sleep-proxy.

- deny-all - This prevents the cache from learning anything. Again incoming on a segment where only clients live. As a result, clients will be able to query for services from the cache (hence the permit 10 match query), but there is no need to learn anything from the clients.
- mixed - This is created to be used in client segments. In addition to clients (such as iPads, PCs, and so on), the occasional printer or a TV will also connect. The purpose here is to learn about those specific services but not about services the clients provide. The filter applied is IN. As a result, the following actions are applicable:
    - Allow every query IN.
    - Allow specific services in (such as printer services [IPP]).
    - Deny everything else.

In addition, to keep the service PTRs fresh in the cache an active query is configured. The active query queries for those services that we want to extend. Typically, this would match the services that have been configured as 'permitted' services in the IN filter. The value is set to 900 seconds. The duration is enough to refresh the PTRs as they typically have a TTL of 4500 seconds.

Example: Creating a Service Instance

```
Device> enable
Device# configure terminal
Device(config)# service-instance mdns-sd service printer-3 regtype _ipp._tcp.local domain
tcp4
Device(config-mdns-sd-si)# ipv4addr 209.165.200.230 255.255.255.0
Device(config-mdns-sd-si)# port 9100
Device(config-mdns-sd-si)# target-hostname fqdn-of-printer.example.com.
Device(config-mdns-sd-si)# txt _ipp._tcp.local=printer3
Device(config-mdns-sd-si)# priority 3
Device(config-mdns-sd-si)# weight 20
Device(config-mdns-sd-si)# exit
```

Note When you create a service-instance, a text record is created even if you do not configure service-instance parameters.

Additional References for Service Discovery Gateway

Related Documents

Related Topic	Document Title
Master Command List	Cisco IOS Master Command List
IP Addressing Services Command Reference	Cisco IOS IP Addressing Services Command Reference

Related Topic	Document Title
Configuring DNS	IP Addressing: DNS Configuration Guide
DNS conceptual information	"Information About DNS" section in IP Addressing: DNS Configuration Guide

Standards and RFCs

Standard/RFC	Title
RFC 6762	Multicast DNS
RFC 6763	DNS-Based Service Discovery
Multicast DNS Internet-Draft	Multicast DNS Internet draft

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Service Discovery Gateway

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for Service Discovery Gateway

Feature Name	Releases	Feature Information
Service Discovery Gateway		The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across L3 boundaries (different subnets). The following commands were introduced or modified: **cache-memory-max, clear mdns cache, clear mdns statistics, debug mdns, match message-type, match service-instance, match service-type, redistribute mdns-sd, service-list mdns-sd, service-policy, service-policy-query, service-routing mdns-sd, show mdns cache, show mdns requests, show mdns statistics**
Service Discovery Gateway—Phase 2		The Service Discovery Gateway feature was enhanced with additional filter and configuration options. The following commands were introduced or modified: **clear mdns cache, clear mdns service-types, clear mdns statistics, designated-gateway, match location, rate-limit, service-instance mdns-sd, service-policy-proximity, service-routing mdns-sd, service-type-enumeration, show mdns cache, show mdns statistics, source-interface**
Service Discovery Gateway—Phase 3		The Service Discovery Gateway feature was enhanced with the following features: • **De-congestion of incoming mDNS traffic using the rate limiting mechanism**—The rate-limit value range was reset to1-100 p/s. • **Redistribution of service-withdrawal announcements across subnets when services are withdrawn, to improve mDNS cache efficiency and to avoid message loops**—The **withdraw-only** option was added to the **redistribute mdns-sd** command. • **A filter criterion for services available and learnt on a specific interface**—The **match learnt-interface** command was added to filter services. • **Enabling and disabling of periodic browsing of services on specific interfaces**—The **service-policy-query (interface)** command was added. For existing, globally configured active queries, the **disable** option was added to disable browsing of services on an interface, retaining the configurations on other interfaces. The following commands were introduced or modified: **match learnt-interface, rate-limit, redistribute mdns-sd, service-policy-query (interface)**