DNS Policies Overview

By robinharwood

Archived: 2026-04-05 17:17:34 UTC

You can use this topic to learn about DNS Policy, which is new in Windows Server 2016. You can use DNS Policy for Geo-Location based traffic management, intelligent DNS responses based on the time of day, to manage a single DNS server configured for split-brain deployment, applying filters on DNS queries, and more. The following items provide more detail about these capabilities.

- **Application Load Balancing.** When you have deployed multiple instances of an application at different locations, you can use DNS policy to balance the traffic load between the different application instances, dynamically allocating the traffic load for the application.
- **Geo-Location Based Traffic Management.** You can use DNS Policy to allow primary and secondary DNS servers to respond to DNS client queries based on the geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the IP address of the closest resource.
- **Split Brain DNS.** With split-brain DNS, DNS records are split into different Zone Scopes on the same DNS server, and DNS clients receive a response based on whether the clients are internal or external clients. You can configure split-brain DNS for Active Directory integrated zones or for zones on standalone DNS servers.
- **Filtering.** You can configure DNS policy to create query filters that are based on criteria that you supply. Query filters in DNS policy allow you to configure the DNS server to respond in a custom manner based on the DNS query and DNS client that sends the DNS query.
- **Forensics.** You can use DNS policy to redirect malicious DNS clients to a non-existent IP address instead of directing them to the computer they are trying to reach.
- **Time of day based redirection.** You can use DNS policy to distribute application traffic across different geographically distributed instances of an application by using DNS policies that are based on the time of day.

New Concepts

In order to create policies to support the scenarios listed above, it is necessary to be able to identify groups of records in a zone, groups of clients on a network, among other elements. These elements are represented by the following new DNS objects:

- **Client subnet:** a client subnet object represents an IPv4 or IPv6 subnet from which queries are submitted to a DNS server. You can create subnets to later define policies to be applied based on what subnet the

requests come from. For instance, in a split brain DNS scenario, the request for resolution for a name such as *www.microsoft.com* can be answered with an internal IP address to clients from internal subnets, and a different IP address to clients in external subnets.

- **Recursion scope:** recursion scopes are unique instances of a group of settings that control recursion on a DNS server. A recursion scope contains a list of forwarders and specifies whether recursion is enabled. A DNS server can have many recursion scopes. DNS server recursion policies allow you to choose a recursion scope for a set of queries. If the DNS server is not authoritative for certain queries, DNS server recursion policies allow you to control how to resolve those queries. You can specify which forwarders to use and whether to use recursion.
- **Zone scopes:** a DNS zone can have multiple zone scopes, with each zone scope containing their own set of DNS records. The same record can be present in multiple scopes, with different IP addresses. Also, zone transfers are done at the zone scope level. That means that records from a zone scope in a primary zone will be transferred to the same zone scope in a secondary zone.

Types of Policy

DNS Policies are divided by level and type. You can use Query Resolution Policies to define how queries are processed, and Zone Transfer Policies to define how zone transfers occur. You can apply Each policy type at the server level or the zone level.

Query Resolution Policies

You can use DNS Query Resolution Policies to specify how incoming resolution queries are handled by a DNS server. Every DNS Query Resolution Policy contains the following elements:

Field	Description	Possible values
Name	Policy name	- Up to 256 characters - Can contain any character valid for a file name
State	Policy state	- Enable (default) - Disabled
Level	Policy level	- Server - Zone
Processing order	Once a query is classified by level and applies on, the server finds the first policy for which the query matches the criteria and applies it to query	- Numeric value - Unique value per policy containing the same level and applies on value

Field	Description	Possible values
Action	Action to be performed by DNS server	- Allow (default for zone level) - Deny (default on server level) - Ignore
Criteria	Policy condition (AND/OR) and list of criterion to be met for policy to be applied	- Condition operator (AND/OR) - List of criteria (see the criterion table below)
Scope	List of zone scopes and weighted values per scope. Weighted values are used for load balancing distribution. For instance, if this list includes datacenter1 with a weight of 3 and datacenter2 with a weight of 5 the server will respond with a record from datacentre1 three times out of eight requests	- List of zone scopes (by name) and weights

Note

Server level policies can only have the values **Deny** or **Ignore** as an action.

The DNS policy criteria field is composed of two elements:

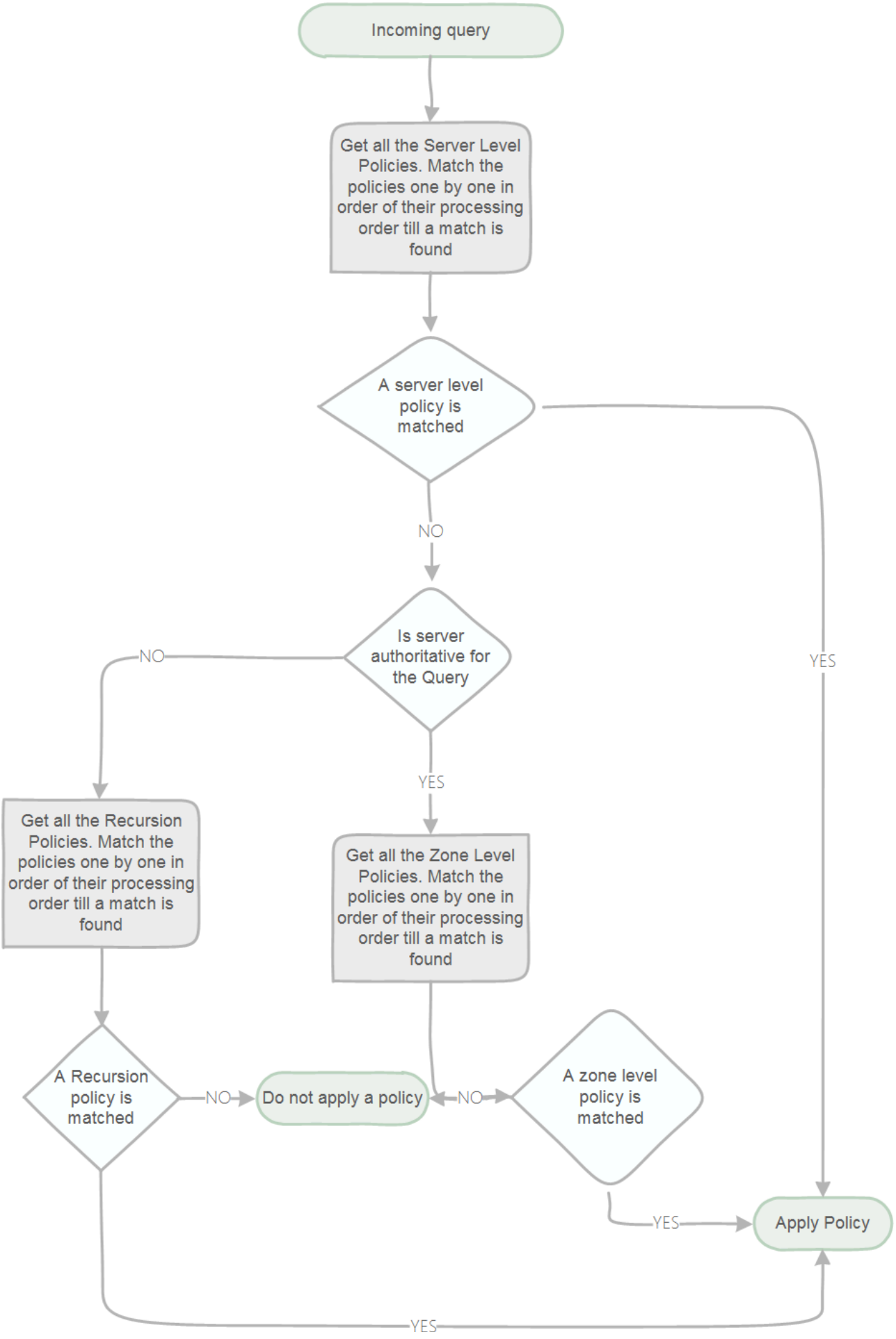
Name	Description	Sample values
Client Subnet	Name of a predefined client subnet. Used to verify the subnet from which the query was sent.	- **EQ,Spain,France** - resolves to true if the subnet is identified as either Spain or France - **NE,Canada,Mexico** - resolves to true if the client subnet is any subnet other than Canada and Mexico
Transport Protocol	Transport protocol used in the query. Possible entries are **UDP** and **TCP**	- **EQ,TCP** - **EQ,UDP**
Internet Protocol	Network protocol used in the query. Possible entries are **IPv4** and **IPv6**	- **EQ,IPv4** - **EQ,IPv6**
Server Interface IP address	IP address for the incoming DNS server network interface	- **EQ,10.0.0.1** - **EQ,192.168.1.1**

Name	Description	Sample values
FQDN	FQDN of record in the query, with the possibility of using a wild card	**- EQ,www.contoso.com -** resolves to true only the if the query is trying to resolve the *www.contoso.com* FQDN **- EQ,*.contoso.com,*.woodgrove.com -** resolves to true if the query is for any record ending in *contoso.com***OR***woodgrove.com*
Query Type	Type of record being queried (A, SRV, TXT)	**- EQ,TXT,SRV** - resolves to true if the query is requesting a TXT **OR** SRV record **- EQ,MX -** resolves to true if the query is requesting an MX record
Time of Day	Time of day the query is received	**- EQ,10:00-12:00,22:00-23:00 -** resolves to true if the query is received between 10 AM and noon, **OR** between 10PM and 11PM

Using the table above as a starting point, the table below could be used to define a criterion that is used to match queries for any type of records but SRV records in the contoso.com domain coming from a client in the 10.0.0.0/24 subnet via TCP between 8 and 10 PM through interface 10.0.0.3:

Name	Value
Client Subnet	EQ,10.0.0.0/24
Transport Protocol	EQ,TCP
Server Interface IP address	EQ,10.0.0.3
FQDN	EQ,*.contoso.com
Query Type	NE,SRV
Time of Day	EQ,20:00-22:00

You can create multiple query resolution policies of the same level, as long as they have a different value for the processing order. When multiple policies are available, the DNS server processes incoming queries in the following manner:


Incoming query
Get all the Server Level Policies. Match the policies one by one in order of their processing order till a match is found
A server level policy is matched
YES
NO
Is server authoritative for the Query
NO
YES
Get all the Recursion Policies. Match the policies one by one in order of their processing order till a match is found
Get all the Zone Level Policies. Match the policies one by one in order of their processing order till a match is found
A Recursion policy is matched
NO
Do not apply a policy
NO
A zone level policy is matched
YES
Apply Policy
YES

Recursion Policies

Recursion policies are a special **type** of server level policies. Recursion policies control how the DNS server performs recursion for a query. Recursion policies apply only when query processing reaches the recursion path. You can choose a value of DENY or IGNORE for recursion for a set of queries. Alternatively, you can choose a set of forwarders for a set of queries.

You can use recursion policies to implement a Split-brain DNS configuration. In this configuration, the DNS server performs recursion for a set of clients for a query, while the DNS server does not perform recursion for other clients for that query.

Recursion policies contains the same elements a regular DNS query resolution policy contains, along with the elements in the table below:

Name	Description
Apply on recursion	Specifies that this policy should only be used for recursion.
Recursion Scope	Name of the recursion scope.

Note

Recursion policies can only be created at the server level.

Zone Transfer Policies

Zone transfer policies control whether a zone transfer is allowed or not by your DNS server. You can create policies for zone transfer at either the server level or the zone level. Server level policies apply on every zone transfer query that occurs on the DNS server. Zone level policies apply only on the queries on a zone hosted on the DNS server. The most common use for zone level policies is to implement blocked or safe lists.

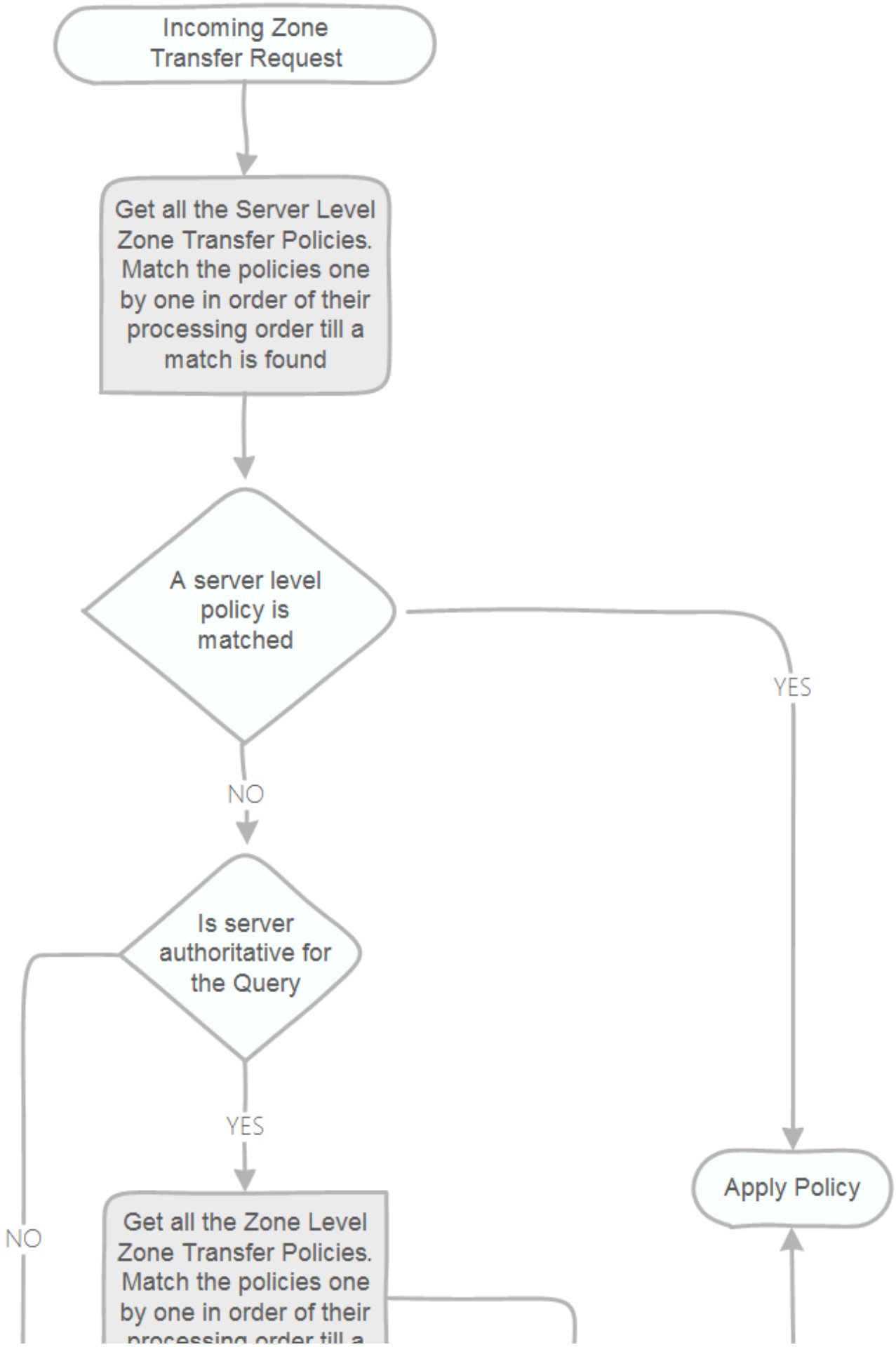
Note

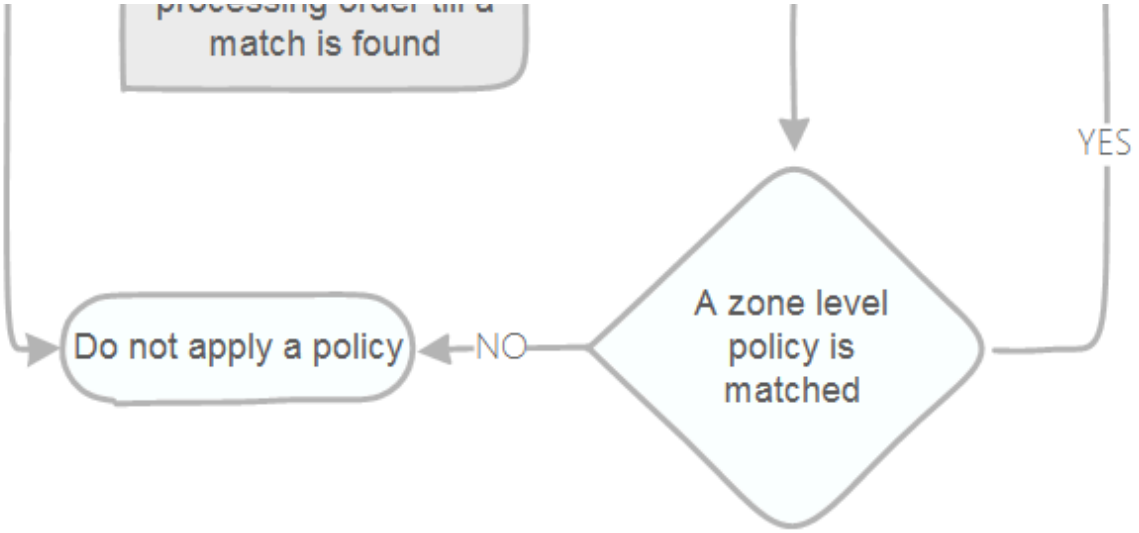
Zone transfer policies can only use DENY or IGNORE as actions.

You can use the server level zone transfer policy below to deny a zone transfer for the contoso.com domain from a given subnet:

```
Add-DnsServerZoneTransferPolicy -Name DenyTransferOfContosoToFabrikam -Zone contoso.com -Action DENY -ClientSu
```

You can create multiple zone transfer policies of the same level, as long as they have a different value for the processing order. When multiple policies are available, the DNS server processes incoming queries in the following manner:


Incoming Zone Transfer Request
Get all the Server Level Zone Transfer Policies. Match the policies one by one in order of their processing order till a match is found
A server level policy is matched
YES
NO
Is server authoritative for the Query
YES
NO
Get all the Zone Level Zone Transfer Policies. Match the policies one by one in order of their
Apply Policy



Managing DNS Policies

You can create and manage DNS Policies by using PowerShell. The examples below go through different sample scenarios that you can configure through DNS Policies:

Traffic Management

You can direct traffic based on an FQDN to different servers depending on the location of the DNS client. The example below shows how to create traffic management policies to direct the customers from a certain subnet to a North American datacenter and from another subnet to a European datacenter.

```
 Add-DnsServerClientSubnet -Name "NorthAmericaSubnet" -IPv4Subnet "172.21.33.0/24"
Add-DnsServerClientSubnet -Name "EuropeSubnet" -IPv4Subnet "172.17.44.0/24"
Add-DnsServerZoneScope -ZoneName "Contoso.com" -Name "NorthAmericaZoneScope"
Add-DnsServerZoneScope -ZoneName "Contoso.com" -Name "EuropeZoneScope"
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "www" -IPv4Address "172.17.97.97" -ZoneScope "Europ
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "www" -IPv4Address "172.21.21.21" -ZoneScope "North
Add-DnsServerQueryResolutionPolicy -Name "NorthAmericaPolicy" -Action ALLOW -ClientSubnet "eq,NorthAmericaSubnet
Add-DnsServerQueryResolutionPolicy -Name "EuropePolicy" -Action ALLOW -ClientSubnet "eq,EuropeSubnet" -ZoneScope
```

The first two lines of the script create client subnet objects for North America and Europe. The two lines after that create a zone scope within the contoso.com domain, one for each region. The two lines after that create a record in each zone that associates `www.contoso.com` to different IP address, one for Europe, another one for North America. Finally, the last lines of the script create two DNS Query Resolution Policies, one to be applied to the North America subnet, another to the Europe subnet.

Block queries for a domain

You can use a DNS Query Resolution Policy to block queries to a domain. The example below blocks all queries to treyresearch.net:

```
Add-DnsServerQueryResolutionPolicy -Name "BlackholePolicy" -Action IGNORE -FQDN "EQ,*.treyresearch.com"
```

Block queries from a subnet

You can also block queries coming from a specific subnet. The script below creates a subnet for 172.0.33.0/24 and then creates a policy to ignore all queries coming from that subnet:

```
 Add-DnsServerClientSubnet -Name "MaliciousSubnet06" -IPv4Subnet 172.0.33.0/24
Add-DnsServerQueryResolutionPolicy -Name "BlackholePolicyMalicious06" -Action IGNORE -ClientSubnet  "EQ,Maliciou
```

Allow recursion for internal clients

You can control recursion by using a DNS Query Resolution Policy. The sample below can be used to enable recursion for internal clients, while disabling it for external clients in a split brain scenario.

```
 Set-DnsServerRecursionScope -Name . -EnableRecursion $False
Add-DnsServerRecursionScope -Name "InternalClients" -EnableRecursion $True
Add-DnsServerQueryResolutionPolicy -Name "SplitBrainPolicy" -Action ALLOW -ApplyOnRecursion -RecursionScope "Int
```

The first line in the script changes the default recursion scope, simply named as "." (dot) to disable recursion. The second line creates a recursion scope named *InternalClients* with recursion enabled. And the third line creates a policy to apply the newly create recursion scope to any queries coming in through a server interface that has 10.0.0.34 as an IP address.

Create a server level zone transfer policy

You can control zone transfer in a more granular form by using DNS Zone Transfer policies. The sample script below can be used to allow zone transfers for any server on a given subnet:

```
 Add-DnsServerClientSubnet -Name "AllowedSubnet" -IPv4Subnet 172.21.33.0/24
Add-DnsServerZoneTransferPolicy -Name "NorthAmericaPolicy" -Action IGNORE -ClientSubnet "ne,AllowedSubnet"
```

The first line in the script creates a subnet object named *AllowedSubnet* with the IP block 172.21.33.0/24. The second line creates a zone transfer policy to allow zone transfers to any DNS server on the subnet previously created.

Create a zone level zone transfer policy

You can also create zone level zone transfer policies. The example below ignores any request for a zone transfer for contoso.com coming in from a server interface that has an IP address of 10.0.0.33:

```
Add-DnsServerZoneTransferPolicy -Name "InternalTransfers" -Action IGNORE -ServerInterfaceIP "eq,10.0.0.33" -Pas
```

DNS Policy Scenarios

For information on how to use DNS policy for specific scenarios, see the following topics in this guide.

- Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers
- Use DNS Policy for Geo-Location Based Traffic Management with Primary-Secondary Deployments
- Use DNS Policy for Intelligent DNS Responses Based on the Time of Day
- DNS Responses Based on Time of Day with an Azure Cloud App Server
- Use DNS Policy for Split-Brain DNS Deployment
- Use DNS Policy for Split-Brain DNS in Active Directory
- Use DNS Policy for Applying Filters on DNS Queries
- Use DNS Policy for Application Load Balancing
- Use DNS Policy for Application Load Balancing With Geo-Location Awareness

Using DNS Policy on Read-Only Domain Controllers

DNS Policy is compatible with Read-Only Domain Controllers. Do note that a restart of the DNS Server service is required for new DNS Policies to be loaded on Read-Only Domain Controllers. This is not necessary on writable domain controllers.
