# BIG-IP® Local Traffic Management: Basics

Version 11.6

# Table of Contents

# Introduction to Local Traffic Management

## About local traffic management

Central to the BIG-IP® local traffic management system are virtual servers, nodes, and server pools. Virtual servers receive incoming traffic, perform basic destination IP address translation, and direct traffic to server nodes, which are grouped together in pools.

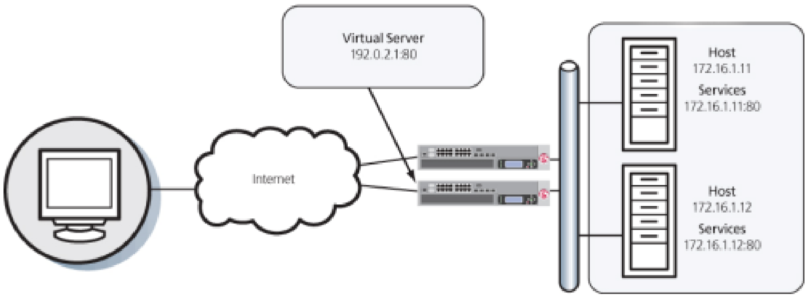


**Figure 1: A basic local traffic management configuration**

To configure a basic local traffic management system, you use the BIG-IP Configuration utility. With this utility, you can create a complete set of virtual servers, nodes, and server pools that work together to perform local traffic management. Each object has a set of configuration settings that you can use as is or change to suit your needs.

## About the network map

The BIG-IP® Configuration utility includes a feature known as the network map. The *network map* shows a summary of local traffic objects, as well as a visual map showing the relationships among the virtual servers, pools, and pool members on the BIG-IP® system. For both the local traffic summary and the visual map, you can customize the display using a search mechanism that filters the information that you want to display, according to criteria that you specify. The system highlights in blue all matches from a search operation.

### Viewing the network map

You perform this task to view the hierarchical relationship of local traffic objects to each virtual server on the BIG-IP® system. The feature includes a filtering mechanism for display only the objects and the status that you want to view.

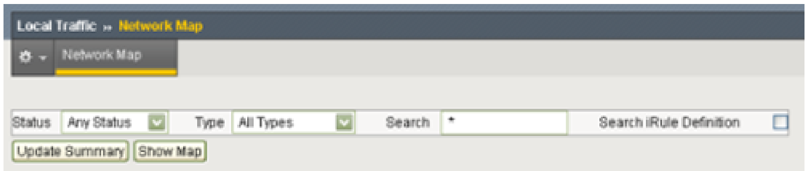
1. On the Main tab, click **Local Traffic** > **Network Map**.
2. View the relationship of objects associated with each virtual server on the BIG-IP system.
3. If you want to filter the objects you see, based on object status:

a) From the **Status** list, select a status.
b) From the **Type** list, select an object type.
c) Click the **Update Map** button.

After performing this task, you can see a network map for each virtual server on the system. The maps display the objects and status the filtering mechanism specifies.

## The filtering mechanism

You can filter the results of the network map feature by using the Type and Status lists in the filter bar, as well as a Search box. With the Search box, you can optionally type a specific string. Figure 1.1 shows the filtering options on the Network Map screen.



Status: Any Status
Type: All Types
Search: *
Search iRule Definition □

**Figure 2: Filtering options on the Network Map screen**

When using the Search box, you can specify a text string that you want the system to use in a search operation. The default is asterisk ( * ). The settings of the Status and Type fields determine the scope of the search. The system uses the specified search string to filter the results that the system displays on the screen.

For example, if you constrain the search to include only unavailable nodes whose IP address includes 10.10, the operation returns those nodes, along with the members of the pool, the pool itself, the associated virtual server, and any iRules® that you explicitly applied to that virtual server. The system sorts results alphabetically, by virtual server name.

The system supports searching on names, IP address, and IP address:port combinations, in both IPv4 and IPv6 address formats. The system processes the string as if an asterisk wildcard character surrounds the string. For example, you specify `10`, the system effectively searches as if you had typed `*10*`. You can also specifically include the asterisk wildcard character. For example, you can use the following search strings: `10.10.10.*:80`, `10.10*`, and `*:80`. if you specifically include a wildcard character, the system treats the string accordingly. For example, if you specify `10*`, the system assumes you want to search for objects whose IP addresses begin with 10.

***Tip:*** *Browsers have limits as to how much data they can render before they become sluggish and halt processing. Mapping large configurations might approach those limits; therefore, memory constraints might prevent the system from producing a network map of the whole configuration. If this might happen, the system posts an alert indicating that you can use the Network Map summary screen to determine the complexity of the configuration, which can give you an indication of the size of the resulting map. You can modify the search criteria to return fewer results, producing a map that does not encounter those limits.*

## Object summary

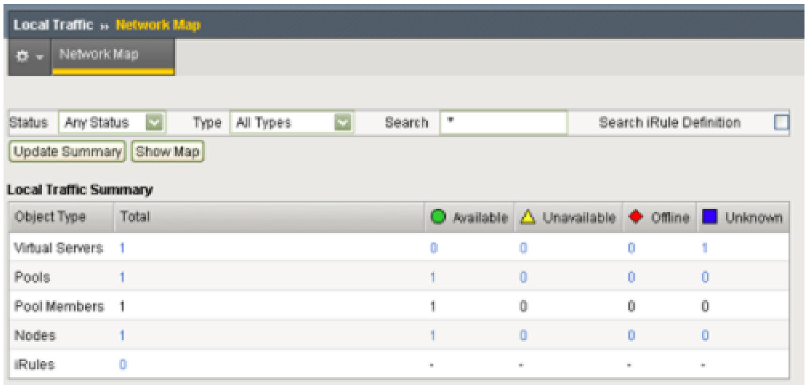
When you first open the Network Map screen, the screen displays a summary of local traffic objects. This summary includes the type of objects specified with the search mechanism, the number of each type of object, and, for each object type, the number of objects with a given status.

The summary displays data for these object types:

- Virtual servers
- Pools
- Pool members
- Nodes
- iRules

***Note:*** *A local traffic summary includes only those objects that are referenced by a virtual server. For example, if you have configured a pool on the system but there is no virtual server that references that pool, the local traffic summary does not include that pool, its members, or the associated nodes in the summary.*

This figure shows an example of a network map screen that summarizes local traffic objects on the system.



Status: Any Status | Type: All Types | Search: * | Search iRule Definition □

Local Traffic Summary

| Object Type | Total | ● Available | △ Unavailable | ◆ Offline | ■ Unknown |
|---|---|---|---|---|---|
| Virtual Servers | 1 | 0 | 0 | 0 | 1 |
| Pools | 1 | 1 | 0 | 0 | 0 |
| Pool Members | 1 | 1 | 0 | 0 | 0 |
| Nodes | 1 | 1 | 0 | 0 | 0 |
| iRules | 0 | - | - | - | - |

**Figure 3: Local Traffic summary**

# About Virtual Servers

## Introduction to virtual servers

A virtual server is one of the most important components of any BIG-IP® system configuration. A *virtual server* is a traffic-management object on the BIG-IP system that is represented by a virtual IP address and a service, such as `192.168.20.10:80`. When clients on an external network send application traffic to virtual server, the virtual server listens for that traffic and, through destination address translation, directs the traffic according to the way that you configured the settings on the virtual server. A primary purpose of a virtual server is to distribute traffic across a pool of servers that you specify in the virtual server configuration.

o customize the way that the BIG-IP system processes various types of traffic, you can assign profiles to a virtual server. For example, through profile assignment, a virtual server can enable compression on HTTP request data as it passes through the BIG-IP system, or decrypt and re-encrypt SSL connections and verify SSL certificates. For each type of traffic, such as TCP, UDP, HTTP, SSL, SIP, and FTP, you can assign a custom profile to the virtual server or use the default profile.

When you create a virtual server, you specify the pool or pools that you want to use as the destination for any traffic coming from that virtual server. You also configure its general properties, profiles, SNATs, and other resources you want to assign to it, such as iRules or session persistence types.

***Note:*** *To ensure that a server response returns through the BIG-IP system, you can either configure the default route on the server to be a self IP address on an internal VLAN, or you can create a SNAT and assign it to a virtual server.*

## Types of virtual servers

You can create several different types of virtual servers, depending on your particular configuration needs.

**Table 1: Types of virtual servers**

| Type | Description |
|---|---|
| Standard | A *Standard* virtual server (also known as a *load balancing* virtual server) directs client traffic to a load balancing pool and is the most basic type of virtual server. When you first create the virtual server, you assign an existing default pool to it. From then on, the virtual server automatically directs traffic to that default pool. |
| Forwarding (Layer 2) | You can set up a *Forwarding (Layer 2)* virtual server to share the same IP address as a node in an associated VLAN. To do this, you must perform some additional configuration tasks. These tasks consist of: creating a VLAN group that includes the VLAN in which the node resides, assigning a self-IP address to the VLAN group, and disabling the virtual server on the relevant VLAN. |
| Forwarding (IP) | A *Forwarding (IP)* virtual server is just like other virtual servers, except that a forwarding virtual server has no pool members to load balance. The virtual server simply forwards the packet directly to the destination IP address specified in the client request. When you |

| Type | Description |
|---|---|
| | use a forwarding virtual server to direct a request to its originally-specified destination IP address, the BIG-IP system adds, tracks, and reaps these connections just as with other virtual servers. You can also view statistics for a forwarding virtual servers. |
| Performance (HTTP) | A *Performance (HTTP)* virtual server is a virtual server with which you associate a Fast HTTP profile. Together, the virtual server and profile increase the speed at which the virtual server processes HTTP requests. |
| Performance (Layer 4) | A *Performance (Layer 4)* virtual server is a virtual server with which you associate a Fast L4 profile. Together, the virtual server and profile increase the speed at which the virtual server processes Layer 4 requests. |
| Stateless | A *stateless* virtual server prevents the BIG-IP system from putting connections into the connection table for wildcard and forwarding destination IP addresses. When creating a stateless virtual server, you cannot configure SNAT automap, iRules, or port translation, and you must configure a default load balancing pool. Note that this type of virtual server applies to UDP traffic only. |
| Reject | A *Reject* virtual server specifies that the BIG-IP system rejects any traffic destined for the virtual server IP address. |
| DHCP | A *DHCP* virtual server relays Dynamic Host Control Protocol (DHCP) messages between clients and servers residing on different IP networks. Known as a *DHCP relay agent*, a BIG-IP system with a DHCP type of virtual server listens for DHCP client messages being broadcast on the subnet and then relays those messages to the DHCP server. The DHCP server then uses the BIG-IP system to send the responses back to the DHCP client. Configuring a DHCP virtual server on the BIG-IP system relieves you of the tasks of installing and running a separate DHCP server on each subnet. |
| Internal | An *internal virtual server* is one that can send traffic to an intermediary server for specialized processing before the standard virtual server sends the traffic to its final destination. For example, if you want the BIG-IP system to perform content adaptation on HTTP requests or responses, you can create an internal virtual server that load balances those requests or responses to a pool of ICAP servers before sending the traffic back to the standard virtual server. An internal virtual server supports both TCP and UDP traffic. |

## Creating a virtual server

Before creating a virtual server, verify that you have created the pool to which you want this virtual server to send traffic.

When you create a virtual server, you specify a destination IP address and service port. All other settings on the virtual server have default values. You can change the default values of any settings to suit your needs.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

**Note:** *The IP address for this field needs to be on the same subnet as the external self-IP address.*

5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. Configure any other settings as needed.
7. If this type of virtual server forwards traffic to a pool, then in the Resources area of the screen, from the **Default Pool** list, select the relevant pool name.

After performing this task, you have a virtual server that listens for application traffic and acts according to the values configured within the virtual server.

## About the destination address

When creating a virtual server, you must specify a destination address. You can specify either a host address or a network address, in either IPv4 or IPv6 format:

- With a host address, a virtual server can listen for client connections that are destined for the host address and then direct them to a server in a server pool. If you do not append a prefix (in CIDR notation) to a host address, the default prefix is `/32`.
- With a network address (specifically, an address whose host bit is set to 0), a virtual server can direct client connections that are destined for an entire range of IP addresses, rather than for a single destination IP address. For example, the virtual server can direct client traffic that is destined for any of the nodes on the `192.168.1.0` network to a specific pool such as `ingress-firewalls`. Or, a virtual server can direct a web connection destined to any address within the subnet `192.168.1.0/24`, to the pool `default_webservers`. If you do not append a prefix (in CIDR notation) to a network address, the default prefix is `/24`.

## About wildcard servers

Besides directing client connections that are destined for a specific network or subnet, a virtual server can also direct client connections that have a specific destination IP address that the virtual server does not recognize, such as a transparent device. This type of virtual server is known as a *wildcard* virtual server. Examples of transparent devices are firewalls, routers, proxy servers, and cache servers.

Wildcard virtual servers are a special type of virtual server that have a network IP address as the specified destination address instead of a host IP address.

When the BIG-IP® system does not find a specific virtual server that matches a client's destination IP address, the BIG-IP system matches the client's destination IP address to a wildcard virtual server, designated by an IP address of `0.0.0.0`. The BIG-IP system then forwards the client's packet to one of the firewalls or routers assigned to that virtual server. Wildcardvirtual servers do not translate the destination IP address of the incoming packet.

### Default and port-specific wildcard virtual servers

There are two kinds of wildcard virtual servers that you can create:

### Default wildcard virtual servers

A *default wildcard virtual server* is a wildcard virtual server that uses port 0 and handles traffic for all services. A wildcard virtual server allows traffic from all external VLANs by default. However, you can specifically disable any VLANs that you do not want the default wildcard virtual server to support. Disabling VLANs for the default wildcard virtual server is done by creating a VLAN disabled list. Note that a VLAN disabled list applies to default wildcard virtual servers only. You cannot create a VLAN disabled list for a wildcard virtual server that is associated with one VLAN only.

### Port-specific wildcard virtual servers

A *port-specific wildcard virtual server* handles traffic for a particular service only, and you define the virtual server using a service name or a port number. You can use port-specific wildcard virtual servers for tracking statistics for a particular type of network traffic, or for routing outgoing traffic, such as HTTP traffic, directly to a cache server rather than a firewall or router.

If you use both a default wildcard virtual server and port-specific wildcard virtual servers, any traffic that does not match either a standard virtual server or one of the port-specific wildcard virtual servers is handled by the default wildcard virtual server.

F5 Networks recommends that when you define transparent nodes that need to handle more than one type of service, such as a firewall or a router, you specify an actual port for the node and turn off port translation for the virtual server.

## About multiple wildcard servers

You can define multiple wildcard virtual servers that run simultaneously. Each wildcard virtual server must be assigned to an individual VLAN, and therefore accepts packets from that VLAN only.

In some configurations, you need to set up a wildcard virtual server on one side of the BIG-IP system to distribute connections across transparent devices. You can create another wildcard virtual server on the other side of the BIG-IP system to forward packets to virtual servers receiving connections from the transparent devices and forwarding them to their destination.

# About virtual addresses

A *virtual address* is the specific node or network IP address with which you associate a virtual server. For example, if a virtual server's destination address and service port are `192.168.20.10:80`, then the IP address `192.168.20.10` is a virtual address.

You can create a many-to-one relationship between virtual servers and a virtual address. For example, you can create the three virtual servers `192.168.20.10:80`, `192.168.20.10:443`, and `192.168.20.10:161` for the same virtual address, `192.168.20.10`.

You cannot explicitly create a virtual address; the BIG-IP system creates a virtual address whenever you create a virtual server, if the virtual address has not already been created. However, you can modify the properties of a virtual address, and you can enable and disable a virtual address. When you disable a virtual address, none of the virtual servers associated with that address can receive incoming network traffic.

When you create a virtual server, BIG-IP® internally associates the virtual address with a MAC address. This in turn causes the BIG-IP® system to respond to Address Resolution Protocol (ARP) requests for the virtual address, and to send gratuitous ARP requests and responses with respect to the virtual address. As an option, you can disable ARP activity for virtual addresses, in the rare case that ARP activity affects system performance. This most likely occurs only when you have a large number of virtual addresses defined on the system.

## About virtual address creation

You create a virtual address indirectly when you create the first virtual server with a destination address that includes the virtual address. You do not explicitly create a virtual address.

For example, if you create a virtual server with a destination address of `192.168.30.22:80`, the BIG-IP® system automatically creates the virtual address `192.168.30.22`.

## Viewing virtual address properties

Using the BIG-IP™ Configuration utility, you can view the properties of an existing virtual address on the BIG-IP system.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.
   This displays the list of virtual addresses.
3. In the Name column, click the name of the relevant virtual address.
   This displays the properties of the virtual address.
4. Click the **Cancel** button.

## Modifying a virtual address

You can modify the properties of a virtual address. For example, you might want to assign a virtual address to a different traffic group, or change the conditions under which the system advertises the virtual address to dynamic routing protocols.

1. On the Main tab, click **Local Traffic** > **Virtual Servers** > **Virtual Address List**.
   The Virtual Address List screen opens.
2. In the Name column, click the virtual address that you want to modify.
   This displays the properties of that virtual address.
3. Modify any property values as needed.
4. Click **Update**.

## Virtual address settings

Lists and describes the configuration settings of a virtual address.

| Property | Description | Default Value |
|---|---|---|
| Name | The name that you assign to the virtual address. This name can match the virtual IP address itself. | No default value |
| Partition / Path | The pathname indicating the partition/folder in which the virtual address resides. | /Common |
| Address | The IP address of the virtual server, excluding the service. | No default value |

| Property | Description | Default Value |
|---|---|---|
| Traffic Group | The traffic group that contains this virtual IP address. | traffic-group-1 or traffic-group-local-only |
| Availability | The availability of the virtual address with respect to service checking. | No default value |
| State | The state of the virtual address, that is, **enabled** or **disabled**. | Enabled |
| Auto Delete | A directive that the system should automatically delete the virtual address with the deletion of the last associated virtual server. When cleared (disabled), this setting specifies that the system should retain the virtual address even when all associated virtual servers have been deleted. | Enabled |
| Advertise Route | The virtual-server conditions for which the BIG-IP system should advertise this virtual address to an advanced routing module. This setting only applies when the **Route Advertisement** setting is enabled (checked). Possible values are:<br>• **When any virtual server is available**<br>• **When all virtual server(s) are available**<br>• **Always** | When any virtual server is available |
| Connection Limit | The number of concurrent connections that the BIG-IP system allows on this virtual address. | 0 |
| ARP | A setting that enables or disables ARP requests for the virtual address. When this setting is disabled, the BIG-IP system ignores ARP requests that other routers send for this virtual address. | Enabled (checked) |
| ICMP Echo | A setting that enables, selectively enables, or disables responses to ICMP echo requests on a per-virtual address basis. When this setting is disabled, the BIG-IP system drops any ICMP echo request packets sent to virtual addresses, including standard statistics and logging. Note that the resulting behavior is affected by the value you configure for the **Advertise Route** setting. | Enabled |
| Route Advertisement | A setting that inserts a route to this virtual address into the kernel routing table so that an advanced routing module can redistribute that route to other routers on the network. | Enabled (checked) |

## About virtual servers and route domain IDs

Whenever you configure the **Source Address** and **Destination Address** settings on a virtual server, the BIG-IP system requires that the route domain IDs match, if route domain IDs are specified. To ensure that this requirement is met, the BIG-IP system enforces specific rules, which vary depending on whether you are modifying an existing virtual server or creating a new virtual server.

**Table 2: Modifying an existing virtual server**

| User action | Result |
|---|---|
| In the destination address, you change an existing route domain ID. | The system automatically changes the route domain ID on the source address to match the new destination route domain ID. |
| In the source address, you change an existing route domain ID. | If the new route domain ID does not match the route domain ID in the destination address, the system displays an error message stating that the two route domain IDs must match. |

**Table 3: Creating a new virtual server**

| User action | Result |
|---|---|
| You specify a destination IP address only,with a route domain ID, and do not specify a source IP address. | The source IP address defaults to `0.0.0.0` and inherits the route domain ID from the destination IP address. |
| You specify both source and destination addresses but no route domain IDs. | The BIG-IP system uses the default route domain. |
| You specify both source and destination addresses and a route domain ID on each of the IP addresses. | The BIG-IP system verifies that both route domain IDs match. Otherwise, the system displays an error message. |
| You specify both source and destination addresses and a route domain ID on one of the addresses, but exclude an ID from the other address. | The system verifies that the specified route domain ID matches the ID of the default route domain. Specifically, when one address lacks an ID, the only valid configuration is one in which the ID specified on the other address is the ID of a default route domain. Otherwise, the system displays an error message. |

## About virtual server and virtual address status

At any time, you can determine the status of a virtual server or virtual address, using the BIG-IP® Configuration utility. You can find this information by displaying the list of virtual servers or virtual addresses and viewing the Status column, or by viewing the **Availability** property of the object.

The BIG-IP Configuration utility indicates status by displaying one of several icons, distinguished by shape and color:

- The shape of the icon indicates the status that the monitor has reported for that node.
- The color of the icon indicates the actual status of the node.

## About clustered multiprocessing

The BIG-IP® system includes a performance feature known as Clustered Multiprocessing™, or CMP®. CMP is a traffic acceleration feature that creates a separate instance of the Traffic Management Microkernel

(TMM) service for each central processing unit (CPU) on the system. When CMP is enabled, the workload is shared equally among all CPUs.

Whenever you create a virtual server, the BIG-IP system automatically enables the CMP feature. When CMP is enabled, all instances of the TMM service process application traffic.

When you view standard performance graphs using the BIG-IP Configuration utility, you can see multiple instances of the TMM service (`tmm0`, `tmm1`, and so on).

When CMP is enabled, be aware that:

- While displaying some statistics individually for each TMM instance, the BIG-IP system displays other statistics as the combined total of all TMM instances.
- Connection limits for a virtual server with CMP enabled are distributed evenly across all instances of the TMM service.

***Note:*** *F5 recommends that you disable the CMP feature if you set a small connection limit on pool members (for example, a connection limit of 2 for the 8400 platform or 4 for the 8800 platform).*

You can enable or disable CMP for a virtual server, or you can enable CMP for a specific CPU.

# About Nodes

## Introduction to nodes

A *node* is a logical object on the BIG-IP® that identifies the IP address or fully-qualified domain name (FQDN) of a physical resource on the network. You can explicitly create a node, or you can instruct the BIG-IP system to automatically create one when you add a pool member to a load balancing pool.

The difference between a node and a pool member is that a node is designated by the device's IP address or FQDN only, while a pool member is designated by an IP address or FQDN and a service (such as `10.10.10.3:80`).

A primary feature of nodes is their association with health monitors. Like pool members, nodes can be associated with health monitors as a way to determine server status. However, a health monitor for a pool member reports the status of a service running on the device, whereas a health monitor for a node reports status of the device itself.

## Creating a node

Local traffic pools use nodes as target resources for load balancing. A node is an IP address or a fully-qualified domain name (FQDN) that represents a server resource that hosts applications.

***Note:*** *An alternate way to create a node is to create a pool member. When you create a pool member, the BIG-IP® system creates the corresponding node for you.*

1. On the Main tab, expand **Local Traffic**, and click **Nodes**.
   The Node List screen opens.
2. Click the **Create** button.
   The New Node screen opens.
3. For the **Address** field:
   a) If you want to specify the node by its IP address, click **Nodes** and type an IP address.
   b) If you want to specify the node by a fully-qualifed domain name (FQDN), click **FQDN** and type the node's FQDN.
4. In the Configuration area of the screen, configure the settings as needed or retain the default values.
5. If you chose **FQDN** for the **Address** setting, then in the FQDN area of the screen, configure the settings as needed or retain the default values.
6. Click **Finished**.
   The screen refreshes, and the new node appears in the node list.

## About the node address setting

This setting specifies the address of the node, either in the form of an IP address or a fully-qualified domain name (FQDN).

If you are using a route domain other than route domain `0`, you can append a route domain ID to any node IP address. For example, if the node address applies to route domain `1`, you can specify a node address of `10.10.10.10%1`.

## About node status

At any time, you can determine the status of a node, using the BIG-IP Configuration utility. You can find this information by displaying the list of nodes and viewing the Status column, or by viewing the Availability property of a node.

The BIG-IP Configuration utility indicates status by displaying one of several icons, distinguished by shape and color:

- The shape of the icon indicates the status that the monitor has reported for that node.
- The color of the icon indicates the actual status of the node.

***Tip:** You can manually set the availability of a node with the Manual Resume attribute of the associated health monitor.*

## About server node state

A node in a server pool must be enabled in order to accept traffic. A *node* is a logical object on the BIG-IP® system that identifies the IP address of a physical resource on the network.

When you disable a node, the BIG-IP® system allows existing connections to time out or end normally. In this case, by default, the only new connections that the node accepts are those that belong to an existing persistence session.

## Additional configuration options

A node object has several optional configuration settings that you can configure.

### About health monitor association

Using the BIG-IP® system, you can monitor the health or performance of your nodes by associating monitors with those nodes. This is similar to associating a monitor with a load balancing pool, except that in the case of nodes, you are monitoring the IP address, whereas with pools, you are monitoring the services that are active on the pool members.

The BIG-IP system contains many different pre-configured monitors that you can associate with nodes, depending on the type of traffic you want to monitor. You can also create your own custom monitors and associate them with nodes. The only pre-configured monitors that are not available for associating with nodes are monitors that are specifically designed to monitor pools or pool members rather than nodes.

***Note:*** *Any monitor that you associate with a node must reside either in partition* `Common` *or in the partition that contains the node.*

There are two ways that you can associate a monitor with a node: by assigning the same monitor (that is, a default monitor) to multiple nodes at the same time, or by explicitly associating a monitor with each node as you create it.

## About monitors and automatic node creation

If you create a pool member without first creating the parent node,the BIG-IP system automatically creates the parent node for you. Fortunately, you can configure the BIG-IP system to automatically associate one or more monitor types with every node that the BIG-IP system creates. This eliminates the task of having to explicitly choose monitors for each node.

Keep the following in mind when working with default monitors:

- If a user with permission to manage objects in partition `Common` disables a monitor that is designated as the default monitor for nodes (such as the `icmp` monitor), this affects all nodes on the system. Ensure that the default monitor for nodes always resides in partition `Common`.
- To specify default monitors, you must have the `Administrator` user role assigned to your user account.
- If all nodes reside in the same partition, the default monitor must reside in that partition or in partition `Common`. If nodes reside in separate partitions, then the default monitor must reside in partition `Common`.

## About monitors and explicit node creation

Sometimes, you might want to explicitly create a node, rather than having Local Traffic Manager™ create the node automatically. In this case, when you create the node, you can either associate non-default monitors with the node, or associate the default monitors with the node.

## About node availability

You can specify the minimum number of health monitors that must report a node as being available to receive traffic before the BIG-IP system reports that node as being in an `up` state.

## About the ratio weight setting

When you are using the Ratio load balancing method, you can assign a ratio weight to each node in a pool. The BIG-IP system uses this ratio weight to determine the correct node for load balancing.

Note that at least one node in the pool must have a ratio value greater than `1`. Otherwise, the effect equals that of the Round Robin load balancing method.

### About the connection rate limit setting

The connection rate limit setting specifies the maximum rate of new connections allowed for the node. When you specify a connection rate limit, the system controls the number of allowed new connections per second, thus providing a manageable increase in connections without compromising availability. The default value of `0` specifies that there is no limit on the number of connections allowed per second.

## Additional FQDN options

A node object has some optional FQDN settings that you can configure.

### About FQDN address types

When you use FQDNs to identify nodes, you must specify whether the FQDN of the node resolves to an IPv4 or IPv6 address.

### About the Auto Populate option

The **Auto Populate** option specifies whether the system automatically creates ephemeral nodes using the IP addresses returned by the resolution of a DNS query for a node defined by an FQDN. The default value is `Enabled`.

When set to `Enabled`, the system generates an ephemeral node for each IP address returned in response to a DNS query for the FQDN of the node. Additionally, when a DNS response indicates the IP address of an ephemeral node no longer exists, the system deletes the ephemeral node.

When set to `Disabled`, the system resolves a DNS query for the FQDN of the node with the single IP address associated with the FQDN.

### About query intervals

You can specify intervals for when a query occurs. The intervals vary depending on whether the DNS server is up or down.

When the DNS server is up, the associated monitor attempts to probe three times, and marks the server down if there is no response within the span of three times the interval value, in seconds. The default value, in seconds, is `3600`. Note that instead of typing an interval, you can enable the Time to Live (**Use TTL**) option.

When the DNS server is down, the associated monitor continues polling as long as the server is down. The default value, in seconds, is `5`.

# About Pools

## Introduction to pools

A *pool* is a logical set of devices, such as web servers, that you group together to receive and process traffic. Instead of sending client traffic to the destination IP address specified in the client request, the BIG-IP® system sends the request to any of the nodes that are members of that pool.

A pool consists of pool members. A *pool member* is a logical object that represents a physical node on the network. Once you have assigned a pool to a virtual server, the BIG-IP system directs traffic coming into the virtual server to a member of that pool. An individual pool member can belong to one or multiple pools, depending on how you want to manage your network traffic.

You can create three types of pools on the system: server pools, gateway pools, and clone pools.

### About server pools

A *server pool* is a pool containing one or more server nodes that process application traffic. The most common type of server pool contains web servers.

One of the properties of a server pool is a load balancing method. A *load balancing method* is an algorithm that the BIG-IP® system uses to select a pool member for processing a request. For example, the default load balancing method is *Round Robin*, which causes the BIG-IP system to send each incoming request to the next available member of the pool, thereby distributing requests evenly across the servers in the pool.

### About gateway pools

One type of pool that you can create is a gateway pool. A *gateway pool* is a pool of routers.

### About clone pools

You use a *clone pool* when you want to configure the BIG-IP system to send traffic to a pool of intrusion detection systems (IDSs). An *intrusion detection system* (IDS) is a device that monitors inbound and outbound network traffic and identifies suspicious patterns that might indicate malicious activities or a network attack. You can use the clone pool feature of a BIG-IP system to copy traffic to a dedicated IDS or a sniffer device.

***Important:*** *A clone pool receives all of the same traffic that the server pool receives.*

To configure a clone pool, you first create a pool of IDS or sniffer devices and then assign the pool as a clone pool to a virtual server. The clone pool feature is the recommended method for copying production traffic to IDS systems or sniffer devices. Note that when you create the clone pool, the service port that you assign to each node is irrelevant; you can choose any service port. Also, when you add a clone pool to a virtual server, the system copies only new connections; existing connections are not copied.

You can configure a virtual server to copy client-side traffic, server-side traffic, or both:

- A *client-side clone pool* causes the virtual server to replicate client-side traffic (prior to address translation) to the specified clone pool.
- *A server-side clone pool* causes the virtual server to replicate server-side traffic (after address translation) to the specified clone pool.

You can configure an unlimited number of clone pools on the BIG-IP system.

## Creating a pool

Before doing this task:

- Decide on the IP addresses or FQDNs for the servers that you want to include in your server pool.
- If your system is using DHCP, make sure your DNS servers aren't configured for round robin DNS resolutions; instead, they should be configured to return all available IP addresses in a resolution.

Use this task to create a pool of servers with pool members. The pool identifies which servers you want the virtual server to send client requests to. As an option, you can identify the servers by their FQDNs instead of their IP addresses. In this way, the system automatically updates pool members whenever you make changes to their corresponding server IP addresses on your network.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select a monitor and move the monitor to the **Active** list.

   ***Note:*** *A pool containing nodes represented by FQDNs cannot be monitored by* ***inband*** *or* ***sasp*** *monitors.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.

6. For the **New Members** setting, add each server that you want to include in the pool:
   a) Select **New Node** or **New FQDN Node**.
   b) (Optional) In the **Node Name** field, type a name for the node.
   c) If you chose **New Node** above, then in the **Address** field, type the IP address of the server. If you chose **New FQDN Node**, then in the **FQDN** field, type the FQDN of the server.

      When using FQDNs, you should still type at least one IP address. Typing one IP address makes sure that the system can find a pool member if a DNS server isn't available.

   d) For the **Service Port** option, pick a service from the list.
   e) If you are using FQDNs for the server names, then for **Auto Populate**, keep the default value of **Enabled**.

      ***Note:*** *When you leave* ***Auto Populate*** *turned on, the system creates an ephemeral node for each IP address returned as an answer to a DNS query. Also, when a DNS answer shows that the IP address of an ephemeral node doesn't exist anymore, the system deletes the ephemeral node.*

   f) Click **Add**.
   g) Do this step again for each node.

7. Click **Finished**.

## Pool and pool member status

An important part of managing pools and pool members is viewing and understanding the status of a pool or pool member at any given time. The BIG-IP Configuration utility indicates status by displaying one of several icons, distinguished by shape and color, for each pool or pool member:

- The shape of the icon indicates the status that the monitor has reported for that pool or pool member. For example, a circle-shaped icon indicates that the monitor has reported the pool member as being `up`, whereas a diamond-shaped icon indicates that the monitor has reported the pool member as being `down`.
- The color of the icon indicates the actual status of the node itself. For example, a green shape indicates that the node is `up`, whereas a red shape indicates that the node is `down`. A black shape indicates that user-intervention is required.

At any time, you can determine the status of a pool. The status of a pool is based solely on the status of its members. Using the BIG-IP Configuration utility, you can find this information by viewing the Availability property of the pool. You can also find this information by displaying the list of pools and checking the Status column.

## Pool features

You can configure the BIG-IP® system to perform a number of different operations for a pool. For example, you can:

- Associate health monitors with pools and pool members
- Enable or disable SNAT connections
- Rebind a connection to a different pool member if the originally-targeted pool member becomes unavailable
- Specify a load balancing algorithm for a pool
- Set the Quality of Service or Type of Service level within a packet
- Assign pool members to priority groups within a pool

You use the BIG-IP Configuration utility to create a load balancing pool, or to modify a pool and its members. When you create a pool, the BIG-IP system automatically assigns a group of default settings to that pool and its members. You can retain these default settings or modify them. Also, you can modify the settings at a later time, after you have created the pool.

### About health monitor association

Health monitors are a key feature of the BIG-IP system. Health monitors help to ensure that a server is in an `up` state and able to receive traffic. When you want to associate a monitor with an entire pool of servers, you do not need to explicitly associate that monitor with each individual server. Instead, you can simply assign the monitor to the pool itself. the BIG-IP system then automatically monitors each member of the pool.

The BIG-IP system contains many different pre-configured monitors that you can associate with pools, depending on the type of traffic you want to monitor. You can also create your own custom monitors and associate them with pools. The only monitor types that are not available for associating with pools are monitors that are specifically designed to monitor nodes and not pools or pool members. That is, the destination address in the monitor specifies an IP address only, rather than an IP address and a service port. These monitor types are:

- ICMP
- TCP Echo
- Real Server
- SNMP DCA
- SNMP DCA Base
- WMI

With the BIG-IP system, you can configure your monitor associations in many useful ways:

- You can associate a health monitor with an entire pool instead of an individual server. In this case, the BIG-IP system automatically associates that monitor with all pool members, including those that you add later. Similarly, when you remove a member from a pool, the BIG-IP system no longer monitors that server.
- When a server that is designated as a pool member allows multiple processes to exist on the same IP address and port, you can check the health or status of each process. To do this, you can add the server to multiple pools, and then within each pool, associate a monitor with the that server. The monitor you associate with each server checks the health of the process running on that server.
- When associating a monitor with an entire pool, you can exclude an individual pool member from being associated with that monitor. In this case, you can associate a different monitor for that particular pool member, or you can exclude that pool member from health monitoring altogether. For example, you can associate pool members `A`, `B`, and `D` with the `http` monitor, while you associate pool member `C` with the `https` monitor.
- You can associate multiple monitors with the same pool. For instance, you can associate both the `http` and `https` monitors with the same pool.

## Pool member availability

You can specify a minimum number of health monitors. Before Local Traffic Manager™ can report the pool member as being in an `up` state, this number of monitors, at a minimum, must report a pool member as being available to receive traffic.

## Secure network address translations (SNATs) and network address translations (NATs)

When configuring a pool, you can specifically disable any secure network address translations (SNATs) or network address translations (NATs) for any connections that use that pool. By default, these settings are enabled. You can change this setting on an existing pool by displaying the Properties screen for that pool.

One case in which you might want to configure a pool to disable SNAT or NAT connections is when you want the pool to disable SNAT or NAT connections for a specific service. In this case, you could create a separate pool to handle all connections for that service, and then disable the SNAT or NAT for that pool.

## Action when a service becomes unavailable

You can specify the action that you want the BIG-IP system to take when the service on a pool member becomes unavailable.

Possible actions are:

- None. This is the default action.
- The BIG-IP® system sends an RST (TCP-only) or ICMP message.
- the BIG-IP system simply cleans up the connection.
- the BIG-IP system selects a different node.

You should configure the system to select a different node in certain cases only, such as:

- When the relevant virtual server is a Performance (Layer 4) virtual server with address translation disabled.
- When the relevant virtual server's Protocol setting is set to `UDP`.
- When the pool is a gateway pool (that is, a pool or routers)

## Slow ramp time

When you take a pool member offline, and then bring it back online, the pool member can become overloaded with connection requests, depending on the load balancing method for the pool. For example, if you use the Least Connections load balancing method, the system sends all new connections to the newly-enabled pool member (because, technically, that member has the least amount of connections).

With the slow ramp time feature, you can specify the number of seconds that the system waits before sending traffic to the newly-enabled pool member. The amount of traffic is based on the ratio of how long the pool member is available compared to the slow ramp time, in seconds. Once the pool member is online for a time greater than the slow ramp time, the pool member receives a full proportion of the incoming traffic.

## Type of Service (ToS) level

Another pool feature is the Type of Service (ToS) level. The *ToS* level is one means by which network equipment can identify and treat traffic differently based on an identifier.

As traffic enters the site, the BIG-IP system can set the ToS level on a packet. Using the IP ToS to Server ToS level that you define for the pool to which the packet is sent. the BIG-IP system can apply an iRule and send the traffic to different pools of servers based on that ToS level.

The BIG-IP system can also tag outbound traffic (that is, the return packets based on an HTTP GET) based on the IP ToS to Client ToS value set in the pool. That value is then inspected by upstream devices and given appropriate priority.

For example, to configure a pool so that a ToS level is set for a packet sent to that pool, you can set both the IP ToS to Client level and the IP ToS to Server levels to `16`. In this case, the ToS level is set to `16` when sending packets to the client and when sending packets to the server.

***Note:*** *If you change the ToS level on a pool for a client or a server, existing connections continue to use the previous setting.*

## Quality of Service (QoS) level

Another setting for a pool is the Quality of Service (QoS) level. In addition to the ToS level, the QoS level is a means by which network equipment can identify and treat traffic differently based on an identifier. Essentially, the QoS level specified in a packet enforces a throughput policy for that packet.

As traffic enters the site, the BIG-IP system can set the QoS level on a packet. Using the Link QoS to Server QoS level that you define for the pool to which the packet is sent, the BIG-IP system can apply an iRule that sends the traffic to different pools of servers based on that QoS level.

The BIG-IP system can also tag outbound traffic (that is, the return packets based on an HTTP GET) based on the Link QoS to Client QoS value set in the pool. That value is then inspected by upstream devices and given appropriate priority.

For example, to configure a pool so that a QoS level is set for a packet sent to that pool, you can set the Link QoS to Client level to 3 and the Link QoS to Server level to 4. In this case, the QoS level is set to 3 when sending packets to the client, and set to 4 when sending packets to the server.

## Number of reselect tries

You can specify the number of times that the system tries to contact a new pool member after a passive failure. A *passive failure* consists of a server-connect failure or a failure to receive a data response within a user-specified interval. The default value of `0` indicates no reselects.

***Note:*** *This setting is for use primarily with TCP profiles. Using this setting with a Fast L4 profile is not recommended.*

## About TCP request queue

*TCP request queuing* provides the ability to queue connection requests that exceed the capacity of connections for a pool, pool member, or node, as determined by the connection limit. Consequently, instead of dropping connection requests that exceed the capacity of a pool, pool member, or node, TCP request queuing enables those connection requests to reside within a queue in accordance with defined conditions until capacity becomes available.

When using session persistence, a request becomes queued when the pool member connection limit is reached.

Without session persistence, when all pool members have a specified connection limit, a request becomes queued when the total number of connection limits for all pool members is reached.

Conditions for queuing connection requests include:

- The maximum number of connection requests within the queue, which equates to the maximum number of connections within the pool, pool member, or node. Specifically, the maximum number of connection requests within the queue cannot exceed the cumulative total number of connections for each pool member or node. Any connection requests that exceed the capacity of the request queue are dropped.
- The availability of server connections for reuse. When a server connection becomes available for reuse, the next available connection request in the queue becomes dequeued, thus allowing additional connection requests to be queued.
- The expiration rate of connection requests within the queue. As queue entries expire, they are removed from the queue, thus allowing additional connection requests to be queued.

Connection requests within the queue become dequeued when:

- The connection limit of the pool is increased.
- A pool member's slow ramp time limit permits a new connection to be made.
- The number of concurrent connections to the virtual server decreases below the connection limit.
- The connection request within the queue expires.

## About load balancing methods

Load balancing is an integral part of the BIG-IP® system. Configuring load balancing on a BIG-IP system means determining your load balancing scenario, that is, which pool member should receive a connection hosted by a particular virtual server. Once you have decided on a load balancing scenario, you can specify the appropriate load balancing method for that scenario.

A *load balancing method* is an algorithm or formula that the BIG-IP system uses to determine the server to which traffic will be sent. Individual load balancing methods take into account one or more dynamic factors, such as current connection count. Because each application of the BIG-IP system is unique, and server performance depends on a number of different factors, we recommend that you experiment with different load balancing methods, and select the one that offers the best performance in your particular environment.

### Default load balancing method

The default load balancing method for the BIG-IP system is Round Robin, which simply passes each new connection request to the next server in line. All other load balancing methods take server capacity and/or status into consideration.

If the equipment that you are load balancing is roughly equal in processing speed and memory, Round Robin method works well in most configurations. If you want to use the Round Robin method, you can skip the remainder of this section, and begin configuring other pool settings that you want to add to the basic pool configuration.

## BIG-IP system load balancing methods

The BIG-IP® system provides several load balancing methods for load balancing traffic to pool members.

| Method | Description | When to use |
|---|---|---|
| Round Robin | This is the default load balancing method. Round Robin method passes each new connection request to the next server in line, eventually distributing connections evenly across the array of machines being load balanced. | Round Robin method works well in most configurations, especially if the equipment that you are load balancing is roughly equal in processing speed and memory. |
| Ratio (member)<br>Ratio (node) | The BIG-IP system distributes connections among pool members or nodes in a static rotation according to ratio weights that you define. In this case, the number of connections that each system receives over time is proportionate to the ratio weight you defined for each pool member or node. You set a ratio weight when you create each pool member or node. | These are static load balancing methods, basing distribution on user-specified ratio weights that are proportional to the capacity of the servers. |
| Dynamic Ratio (member)<br>Dynamic Ratio (node) | The Dynamic Ratio methods select a server based on various aspects of real-time server performance analysis. These methods are similar to the Ratio methods, except that with Dynamic Ratio methods, the ratio weights are system-generated, and the values of the ratio weights are not static. These methods are based on continuous monitoring of the servers, and the ratio weights are therefore continually changing.<br>**Note:** *To implement Dynamic Ratio load balancing, you must first install and configure the necessary server software for these systems, and then install the appropriate performance monitor.* | The Dynamic Ratio methods are used specifically for load balancing traffic to RealNetworks® RealSystem® Server platforms, Windows® platforms equipped with Windows Management Instrumentation (WMI), or any server equipped with an SNMP agent such as the UC Davis SNMP agent or Windows 2000 Server SNMP agent. |
| Fastest (node)<br>Fastest (application) | The Fastest methods select a server based on the least number of current sessions. These methods require that you assign both a Layer 7 and a TCP type of profile to the virtual server.<br>**Note:** *If the OneConnect™ feature is enabled, the Least Connections methods do not include idle connections in the calculations when selecting a pool member or node. The Least* | The Fastest methods are useful in environments where nodes are distributed across separate logical networks. |

| Method | Description | When to use |
|---|---|---|
| | *Connections methods use only active connections in their calculations.* | |
| Least Connections (member) Least Connections (node) | The Least Connections methods are relatively simple in that the BIG-IP system passes a new connection to the pool member or node that has the least number of active connections.<br>***Note:*** *If the OneConnect feature is enabled, the Least Connections methods do not include idle connections in the calculations when selecting a pool member or node. The Least Connections methods use only active connections in their calculations.* | The Least Connections methods function best in environments where the servers have similar capabilities. Otherwise, some amount of latency can occur. For example, consider the case where a pool has two servers of differing capacities, A and B. Server A has 95 active connections with a connection limit of 100, while server B has 96 active connections with a much larger connection limit of 500. In this case, the Least Connections method selects server A, the server with the lowest number of active connections, even though the server is close to reaching capacity. If you have servers with varying capacities, consider using the Weighted Least Connections methods instead. |
| Weighted Least Connections (member) Weighted Least Connections (node) | Similar to the Least Connections methods, these load balancing methods select pool members or nodes based on the number of active connections. However, the Weighted Least Connections methods also base their selections on server capacity. The Weighted Least Connections (member) method specifies that the system uses the value you specify in Connection Limit to establish a proportional algorithm for each pool member. The system bases the load balancing decision on that proportion and the number of current connections to that pool member. For example, member_a has 20 connections and its connection limit is 100, so it is at 20% of capacity. Similarly, member_b has 20 connections and its connection limit is 200, so it is at 10% of capacity. In this case, the system select selects member_b. This algorithm requires all pool members to have a non-zero connection limit specified. The Weighted Least Connections (node) method specifies that the system uses the value you specify in the node's Connection Limit setting and the number of current connections to a node to establish a proportional algorithm. This algorithm requires all nodes used by pool members to have a non-zero connection limit specified. If all servers have equal capacity, these load balancing methods behave in the same way as the Least Connections methods.<br>***Note:*** *If the OneConnect feature is enabled, the Weighted Least Connections methods do not include idle connections in the calculations when selecting a pool member or node. The Weighted Least Connections methods use only active connections in their calculations.* | Weighted Least Connections methods work best in environments where the servers have differing capacities. For example, if two servers have the same number of active connections but one server has more capacity than the other, the BIG-IP system calculates the percentage of capacity being used on each server and uses that percentage in its calculations. |
| Observed (member) Observed (node) | With the Observed methods, nodes are ranked based on the number of connections. The Observed methods track the number of Layer 4 connections to each node over time and create a ratio for load balancing. | The need for the Observed methods is rare, and they are not recommended for large pools. |

| Method | Description | When to use |
|---|---|---|
| Predictive (member)<br>Predictive (node) | The Predictive methods use the ranking methods used by the Observed methods, where servers are rated according to the number of current connections. However, with the Predictive methods, the BIG-IP system analyzes the trend of the ranking over time, determining whether a node's performance is currently improving or declining. The servers with performance rankings that are currently improving, rather than declining, receive a higher proportion of the connections. | The need for the Predictive methods is rare, and they are not recommend for large pools. |
| Least Sessions | The Least Sessions method selects the server that currently has the least number of entries in the persistence table. Use of this load balancing method requires that the virtual server reference a type of profile that tracks persistence connections, such as the Source Address Affinity or Universal profile type.<br>***Note:*** *The Least Sessions methods are incompatible with cookie persistence.* | The Least Sessions method works best in environments where the servers or other equipment that you are load balancing have similar capabilities. |
| Ratio Least Connections | The Ratio Least Connections methods cause the system to select the pool member according to the ratio of the number of connections that each pool member has active. | |

## About priority-based member activation

*Priority-based member activation* is a feature that allows you to categorize pool members into priority groups, so that pool members in higher priority groups accept traffic before pool members in lower priority groups. The priority-based member activation feature has two configuration settings:

### Priority group activation

For the priority group activation setting, you specify the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group. The allowed value for this setting ranges from `0` to `65535`. Setting this value to **0** disables the feature (equivalent to using the default value of **Disabled**).

### Priority group

When you enable priority group activation, you also specify a priority group for each member when you add that member to the pool. Retaining the default priority group value of `0` for a pool member means that the pool member is in the lowest priority group and only receives traffic when all pool members in higher priority groups are unavailable.

If the number of available members assigned to the highest priority group drops below the number that you specify, the BIG-IP® system distributes traffic to the next highest priority group, and so on.

For example, this configuration has three priority groups, `3`, `2`, and `1`, with the priority group activation value (shown here as `min active members`) set to `2`.

```
pool my_pool {
   lb_mode fastest
   min active members 2
   member 10.12.10.7:80 priority 3
   member 10.12.10.8:80 priority 3
   member 10.12.10.9:80 priority 3
   member 10.12.10.4:80 priority 2
   member 10.12.10.5:80 priority 2
```

```
    member 10.12.10.6:80 priority 2
    member 10.12.10.1:80 priority 1
    member 10.12.10.2:80 priority 1
    member 10.12.10.3:80 priority 1
    }
```

Connections are first distributed to all pool members with priority `3` (the highest priority group). If fewer than two priority `3` members are available, traffic is directed to the priority `2` members as well. If both the priority `3` group and the priority `2` group have fewer than two members available, traffic is directed to the priority `1` group. The BIG-IP system continuously monitors the priority groups, and whenever a higher priority group once again has the minimum number of available members, the BIG-IP system limits traffic to that group.

## Pool member features

A pool member consists of a server's IP address and service port number. An example of a pool member is `10.10.10.1:80`. Pool members have a number of features that you can configure when you create the pool.

***Note:** By design, a pool and its members always reside in the same administrative partition.*

### About ratio weights

When using a ratio-based load balancing method for distributing traffic to servers within a pool, you can assign a ratio weight to the corresponding pool members. The ratio weight determines the amount of traffic that the pool member receives. The ratio-based load balancing methods are: Ratio (node, member, and sessions), Dynamic Ratio (node and member), and Ratio Least Connections (node and member).

### About priority group numbers

Using the priority group feature, you can assign a priority number to the pool member. The BIG-IP® system then distributes traffic in the pool according to the priority number that you assigned to the pool member.

For example, pool members assigned to group 3, instead of pool members in group 2 or group 1, normally receive all traffic. Thus, members that are assigned a high priority receive all traffic until the load reaches a certain level or some number of members in the group become unavailable. If either of these events occurs, some of the traffic goes to members assigned to the next higher priority group.

This setting is used in tandem with the pool feature known as priority group activation. You use the *priority group activation* feature to configure the minimum number of members that must be available before the BIG-IP system begins directing traffic to members in a lower priority group.

### About connection limits

**Connection limits**

You can specify the maximum number of concurrent connections allowed for a pool member. Note that the default value of `0` (zero) means that there is no limit to the number of concurrent connections that the pool member can receive.

### Connection rate limits

The maximum rate of new connections allowed for the pool member. When you specify a connection rate limit, the system controls the number of allowed new connections per second, thus providing a manageable increase in connections without compromising availability. The default value of `0` specifies that there is no limit on the number of connections allowed per second. The optimal value to specify for a pool member is between `300` and `5000` connections. The maximum valued allowed is `100000`.

## About health monitors

### Explicit monitor associations

After you have associated a monitor with a pool, the BIG-IP system automatically associates that monitor with every pool member, including those members that you add to the pool later. However, in some cases you might want the monitor for a specific pool member to be different from that assigned to the pool. In this case, you must specify that you want to explicitly associate a specific monitor with the individual pool member. You can also prevent the BIG-IP system from associating any monitor with that pool member.

### Explicit monitor association for a pool member

The BIG-IP system contains many different monitors that you can associate with a pool member, depending on the type of traffic you want to monitor. You can also create your own custom monitors and associate them with pool members. The only monitor types that are not available for associating with pool members are monitors that are specifically designed to monitor nodes and not pools or pool members. These monitor types are:

- ICMP
- TCP Echo
- Real Server
- SNMP DCA
- SNMP DCA Base
- WMI

### Multiple monitor association for a pool member

The BIG-IP system allows you to associate more than one monitor with the same pool member. You can:

- Associate more than one monitor with a member of a single pool. For example, you can create monitors `http1`, `http2`, and `http3`, where each monitor is configured differently, and associate all three monitors with the same pool member. In this case, the pool member is marked as `down` if any of the checks is unsuccessful.
- Assign a single IP address and service to be a member of multiple pools. Then, within each pool, you can associate a different monitor with that pool member. For example, suppose you add the pool member `10.10.10.20:80` to three separate pools: `my_pool1`, `my_pool2`, and `my_pool3`. You can then associate all three HTTP monitors to that same pool member. The result is that the BIG-IP system uses the `http1` monitor to check the health of pool member `10.10.10.20:80` in `my_pool1`, the `http2` monitor to check the health of pool member `10.10.10.20:80` in `my_pool2`, and the `http3` monitor to check the health of pool member `10.10.10.20:80` in `my_pool3`.

You can make multiple-monitor associations either at the time you add the pool member to each pool, or by modifying a pool member's properties later.

### Availability requirement

You can specify a minimum number of health monitors. Before the BIG-IP system can report the pool member as being in an `up` state, this number of monitors, at a minimum, must report a pool member as being available to receive traffic.

## About pool member state

You can enable or disable individual pool members. A *pool member* is a logical object on the BIG-IP® system that represents a specific server node and service. For example, a node with an IP address of `12.10.10.3` can have a corresponding pool member `12.10.10.3:80`.

When you disable a pool member, the node continues to process any active connections or any connections for the current persistence session.

# Enabling and Disabling Local Traffic Objects

## Introduction to local traffic operations

Using the BIG-IP® Configuration utility, you can manage the availability of server resources on the network by enabling and disabling certain local traffic server objects. These objects consist of server nodes and pool members, as well as virtual servers and their associated virtual addresses.

## About server node state

A node in a server pool must be enabled in order to accept traffic. A *node* is a logical object on the BIG-IP® system that identifies the IP address of a physical resource on the network.

When you disable a node, the BIG-IP® system allows existing connections to time out or end normally. In this case, by default, the only new connections that the node accepts are those that belong to an existing persistence session.

### Viewing the state of a node

It is easy to determine whether a node is currently enabled or disabled.

1. On the Main tab, click **Local Traffic** > **Nodes**.
   The Node List screen opens.
2. In the Name column, click a node name.
   This displays the properties of the node.
3. Locate the **State** property and view the selected value.
4. Click **Cancel**.

### Enabling a node

You can enable a local traffic node that is currently disabled. When you enable a node, the BIG-IP® system allows all types of connections, including persistent connections.

1. On the Main tab, click **Local Traffic** > **Nodes**.
   The Node List screen opens.
2. In the Name column, locate the node you want to enable.
3. Select the check box to the left of the node name.
4. Click the **Enable** button.

After you perform this task, the selected node is available to process application traffic.

### Disabling a node except for persistent/active connections

You perform this task to disable a local traffic node that is currently enabled. When you disable a node, the BIG-IP® system disallows any incoming connections, but continues to process any persistent or active connections.

1. On the Main tab, click **Local Traffic** > **Nodes**.
   The Node List screen opens.
2. In the Name column, locate the node you want to disable.
3. Select the check box to the left of the node name.
4. Click the **Disable** button.

After you perform this task, the selected node is disabled.

### Forcing a node with active connections offline

You can disable a local traffic node to disallow all connections except for active connections. When you disable the node, the BIG-IP® system disallows any incoming connections (including persistent connections), but continues to process active connections. When all active connections have been processed, the node is fully offline.

1. On the Main tab, click **Local Traffic** > **Nodes**.
   The Node List screen opens.
2. In the Name column, click a node name.
   This displays the properties of the node.
3. For the **State** property, click **Forced Offline (Only active connections allowed)**.
4. At the bottom of the screen, click **Update**.

After you perform this task, the selected node will be offline when all active connections have finished processing.

## About pool member state

You can enable or disable individual pool members. A *pool member* is a logical object on the BIG-IP® system that represents a specific server node and service. For example, a node with an IP address of `12.10.10.3` can have a corresponding pool member `12.10.10.3:80`.

When you disable a pool member, the node continues to process any active connections or any connections for the current persistence session.

### Viewing the state of a pool member

Before performing this task, determine the pool member that you want to force offline.

You can force a pool member, while allowing active connections to be completed before the BIG-IP system takes the member offline.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click a pool name in the Pool List.
3. On the menu bar, click **Members**.
4. In the Member list, select the relevant pool member.
5. Locate the **State** property and view the selected value.
6. Click **Cancel**.

## Enabling a pool member

Before performing this task, verify that the corresponding server node address is enabled.

You can enable a pool member that was previously disabled and is currently in an Offline state. You typically enable a pool member when you want to allow the associated virtual server to send traffic to that service on the server node. For example, if you want to allow the BIG-IP system to resume sending traffic to the `http` service on node `12.10.10.3`, you can enable pool member `12.10.10.3:80`.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click a pool name in the Pool List.
3. On the menu bar, click **Members**.
4. Locate the pool member you want to enable and select the check box to the left of the member name.
5. Click **Enable**.

After you perform this task, the service on the associated server node is available for processing traffic.

## Disabling a pool member except for persistent/active connections

Before performing this task, determine the pool member that you want to disable.

You can disable a pool member that was previously enabled and is currently in an Available state. You typically disable a pool member when you want to prevent the associated virtual server from sending traffic to that service on the server node. For example, if you want to prevent the BIG-IP system from sending traffic to the `http` service on node `12.10.10.3`, you can disable pool member `12.10.10.3:80`.

When you perform this task, the BIG-IP system, by default, allows persistent and active connections to be completed before the BIG-IP system marks the pool member as Offline.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click a pool name in the Pool List.
3. On the menu bar, click **Members**.
4. In the Name column, locate the pool member you want to disable.
5. Select the check box to the left of the pool member name.
6. Click the **Disable** button.

After you perform this task, the service on the relevant node is unavailable for processing traffic, except for persistent and active connections.

### Forcing a pool member with active connections offline

Before performing this task, determine the pool member that you want to force offline.

You can force a pool member to accept no new connections, while allowing active connections to be completed before the BIG-IP system takes the member offline.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click a pool name in the Pool List.
3. On the menu bar, click **Members**.
4. In the Member list, select the relevant pool member.
5. For the **State** property, click **Forced Offline (Only active connections allowed)**.
6. Click **Update**.
   The screen refreshes, and the status in the Availability area changes.

## About virtual address state

You can enable or disable virtual addresses to manage virtual server availability to process traffic. You typically disable a virtual address when you want to drop or redirect traffic destined for all virtual servers associated with that virtual address.

### Enabling a virtual address

You can enable a virtual address that is currently disabled. When you enable a virtual address, all associated virtual servers listen for traffic destined for that virtual address and process the traffic accordingly.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.
   This displays the list of virtual addresses.
3. In the Name column, locate the virtual address you want to enable.
4. Select the check box to the left of the virtual address name.
5. Click the **Enable** button.

After performing this task, you can enable any virtual servers corresponding to this virtual address.

### Disabling a virtual address

You can disable a virtual address that is currently enabled. When disabled, all associated virtual servers no longer listen for traffic destined for that virtual address.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.

This displays the list of virtual addresses.

3. In the Name column, locate the virtual address you want to disable.
4. Select the check box to the left of the virtual address name.
5. Click the **Disable** button.

### Viewing the state of a virtual address

You can view the state of a virtual address in preparation for managing virtual server availability to process traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.
   This displays the list of virtual addresses.
3. In the Name column, locate the relevant virtual address and in the State column, view its state.

## About virtual server state

You can enable or disable virtual servers to manage virtual server availability to process traffic. You typically disable a virtual server when you want to drop or redirect traffic destined for a specific IP address and service.

### Viewing the state of a virtual server

You view the state of a virtual server to determine whether a virtual server is currently enabled or disabled.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. In the Name column, click the name of the relevant virtual server.
   This displays the properties of the virtual server.
3. Locate the **State** property and view the selected value.
4. Click the **Cancel** button.

### Enabling a virtual server

Before enabling a virtual server, verify that the corresponding virtual address is enabled.

You perform this task to enable a virtual server that is currently disabled. When enabled, the virtual server listens for traffic destined for the virtual server's IP address and service and processes the traffic according to the virtual server configuration.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. In the Name column, locate the virtual server you want to enable.

3. Select the check box to the left of the virtual server name.
4. Click the **Enable** button.

After you perform this task, the virtual server listens for application traffic destined for both the virtual server IP address and service, and then processes the traffic accordingly.

## Disabling a virtual server

You perform this task when you want to disable a virtual server that is currently enabled. When disabled, the virtual server no longer listens for traffic destined for the IP address and port specified in the virtual server configuration.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. In the Name column, locate the virtual server you want to disable.
3. Select the check box to the left of the virtual server name.
4. Click the **Disable** button.

# Local Traffic Policies

## About local traffic policy matching

BIG-IP® *local traffic policies* comprise a prioritized list of rules that match defined conditions and run specific actions, which you can assign to a virtual server that directs traffic accordingly. For example, you might create a policy that determines whether a client's browser is a Chrome browser and adds an `Alternative-Protocols` attribute to the header, so that subsequent requests from the Chrome browser are directed to a SPDY virtual server. Or you might create a policy that determines whether a client is using a mobile device, and then redirects its requests to the applicable mobile web site's URL.

### Creating a user-defined local traffic policy

You can use BIG-IP® local traffic policy matching to direct traffic in accordance with rules, which are applied as determined by the specified strategy, conditions, and actions.

1. On the Main tab, click **Local Traffic** > **Policies** > **Policy List**.
   The Policy List screen opens.
2. Click **Create**.
   The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.
4. From the **Strategy** list, select a matching strategy.
5. For the **Requires** setting, select a protocol entry from the **Available** list, and move the entry to the **Selected** list using the Move button.
6. For the **Controls** setting, select a functional area or module from the **Available** list, and move the entry to the **Selected** list using the Move button.
7. Click **Add**.
   The New Rule screen opens.
8. In the **Rule** field, type a unique name for the rule.
9. Using the **Conditions** setting, configure the applicable options.
   a) From the **Operand** list, select an operand.
   b) From the **Event** list, select an event.
   c) From the **Selector** list, select the applicable setting.
   d) Select the **Negate** check box to reverse the policy conditions.
   e) From the **Condition** list, select a condition.
   f) Select the **case sensitive** check box to apply case sensitivity to the condition.
   g) In the **Values** field, type the text that applies to the condition and click **Add**.

      The condition text value appears in the **Values** list box.
   h) To the left, near the **Missing** setting, click **Add**.

      The configured condition appears in the **Conditions** list.
10. Using the **Actions** setting, configure the applicable options.

a) From the **Target** list, select a target.
b) From the **Event** list, select an event.
c) From the **Action** list, select an action.
d) From the **Parameters** list, select a type of parameter to apply.
e) In the **Parameters** field, type the text that applies to the type of parameter and click **Add**.

   The configured parameter appears in the **Parameters** list box.
f) At the lower left, click **Add**.

   The configured settings for the action appear in the **Actions** list.

**11.** Click **Finished**.

The policy appears in the list on the Policies List screen.

## About strategies for local traffic policy matching

Each BIG-IP® local traffic matching policy requires a matching strategy to determine the rule that applies if more than one rule matches.

The BIG-IP policies provide three policy matching strategies: a first-match, best-match, and all-match strategy. Each policy matching strategy prioritizes rules according to the rule's position within the Rules list.

***Note:*** *A rule without conditions becomes the default rule in a best-match or first-match strategy, when the rule is the last entry in the Rules list.*

### Table 4: Policy matching strategies

| Matching strategy | Description |
|---|---|
| First-match strategy | A *first-match strategy* starts the actions for the first rule in the Rules list that matches. |
| Best-match strategy | A *best-match strategy* selects and starts the actions of the rule in the Rules list with the best match, as determined by the following factors.<br>• The number of conditions and operands that match the rule.<br>• The length of the matched value for the rule.<br>• The priority of the operands for the rule.<br>***Note:*** *In a best-match strategy, when multiple rules match and specify an action, conflicting or otherwise, only the action of the best-match rule is implemented. A best-match rule can be the lowest ordinal, the highest priority, or the first rule that matches in the Rules list.* |
| All-match strategy | An *all-match strategy* starts the actions for all rules in the Rules list that match.<br>***Note:*** *In an all-match strategy, when multiple rules match, but specify conflicting actions, only the action of the best-match rule is implemented. A best-match rule can be the lowest ordinal, the highest priority, or the first rule that matches in the Rules list.* |

## Local traffic policy matching Requires profile settings

This table summarizes the profile settings that are required for local traffic policy matching.

| Requires Setting | Description |
|---|---|
| **http** | Specifies that the policy matching requires an HTTP profile. |
| **ssl** | Specifies that the policy matching requires a Client SSL profile. |
| **tcp** | Specifies that the policy matching requires a TCP profile. |

## Local traffic policy matching Controls settings

This table summarizes the controls settings that are required for local traffic policy matching.

| Controls Setting | Description |
|---|---|
| **acceleration** | Provides controls associated with acceleration functionality. |
| **caching** | Provides controls associated with caching functionality. |
| **classification** | Provides controls associated with classification. |
| **compression** | Provides controls associated with HTTP compression. |
| **forwarding** | Provides controls associated with forwarding functionality. |
| **request-adaptation** | Provides controls associated with request-adaptation functionality. |
| **response-adaptation** | Provides controls associated with response-adaptation functionality. |
| **server-ssl** | Provides controls associated with server-ssl functionality. |

# About rules for local traffic policy matching

BIG-IP® local traffic policy *rules* match defined conditions and start specific actions. You can create a policy with rules that are as simple or complex as necessary, based on the passing traffic. For example, a rule might simply determine that a client's browser is a Chrome browser that is not on an administrator network. Or a rule might determine that a request URL starts with `/video`, that the client is a mobile device, and that the client's subnet does not match `172.27.56.0/24`.

# About conditions for local traffic policy matching

The *conditions* for a local traffic policy rule define the necessary criteria that must be met in order for the rule's actions to be applied. For example, a policy might include the following conditions, which, when met by a request, would allow the rule's specified actions to be applied.

| Condition | Setting |
|---|---|
| **Operand** | **http-host** |
| **Event** | **request** |

| Condition | Setting |
|---|---|
| **Selector** | **all** |
| **Condition** | **equals** |
| **Values** | `www.siterequest.com` |

## Local traffic policy matching Conditions operands

This table summarizes the operands for each condition used in policy matching.

| Operand | Type | Valid Events | Selectors and Parameters | Description |
|---|---|---|---|---|
| **client-ssl** | string/number | • **request**<br>• **response** | • **cipher**<br>• **cipher-bits**<br>• **protocol** | Requires a Client SSL profile for policy matching. |
| **http-basic-auth** | string | • **request** | • **password**<br>• **username** | Returns <username>:<password> or parts of it. |
| **http-cookie** | string | • **request** | • **all**<br>  • **name** | Returns the value of a particular cookie or cookie attribute. |
| **http-header** | string | • **request**<br>• **response** | • **all**<br>  • **name** (required) | Returns the value of a particular header. |
| **http-host** | string/number | • **request** | • **all**<br>• **host**<br>• **port** | Provides all or part of the HTTP Host header. |
| **http-method** | string | • **request** | • **all** | Provides the HTTP method. |
| **http-referer** | string/number | • **request** | • **all**<br>• **extension**<br>• **host**<br>• **path**<br>• **path-segment**<br>  • **index** (required)<br>• **port**<br>• **query-parameter**<br>  • **name** (required)<br>• **query-string**<br>• **scheme**<br>• **unnamed-query-parameter**<br>  • **index** (required) | Provides all or part of the HTTP Referer header. |

| Operand | Type | Valid Events | Selectors and Parameters | Description |
|---|---|---|---|---|
| **http-set-cookie** | string | • **response** | • **domain**<br>  • **name** (required)<br>• **expiry**<br>  • **name** (required)<br>• **path**<br>  • **name** (required)<br>• **value**<br>  • **name** (required)<br>• **version**<br>  • **name** (required) | Sets the selected setting of a particular cookie or cookie attribute. |
| **http-status** | string/number | • **response** | • **all**<br>• **code**<br>• **text** | Returns the HTTP status line or part of it. |
| **http-uri** | string/number | • **request** | • **all**<br>• **extension**<br>• **host**<br>• **path**<br>• **path-segment**<br>  • **index** (required)<br>• **port**<br>• **query-parameter**<br>  • **name** (required)<br>• **query-string**<br>• **scheme**<br>• **unnamed-query-parameter**<br>  • **index** (required) | Provides all or part of the request URI. |
| **http-version** | string/number | • **request**<br>• **response** | • **response**<br>  • **all**<br>  • **major**<br>  • **minor**<br>  • **protocol** | Provides HTTP/1.1 a number. |
| **tcp** | number | • **request**<br>• **response** | • **address**<br>  • **internal true**<br>  • **local true**<br>• **mss**<br>  • **internal true** | Requires a TCP profile for policy matching. |

| Operand | Type | Valid Events | Selectors and Parameters | Description |
|---|---|---|---|---|
| | | | • **port**<br>• **internal true**<br>• **local true**<br>• **route-domain**<br>• **internal true**<br>• **rtt**<br>• **internal true**<br>• **vlan**<br>• **internal true**<br>• **vlan-id**<br>• **internal true** | |

## About actions for a local traffic policy rule

The *actions* for a local traffic policy rule determine how traffic is handled. For example, actions for a rule could include the following ways of handling traffic.

- Blocking traffic
- Rewriting a URL
- Logging traffic
- Adding a specific header
- Redirecting traffic to a different pool member
- Selecting a specific Web Application policy

### Local traffic policy matching Actions operands

This table summarizes the actions associated with the conditions of the rule used in policy matching.

| Target | Type | Valid Events | Action |
|---|---|---|---|
| **acceleration** | string/number | • **request** | • **disable**<br>• **enable** |
| **cache** | string | • **request**<br>• **response** | • **disable**<br>• **enable**<br>• **pin true** |
| **compress** | string | • **request**<br>• **response** | • **disable**<br>• **enable** |
| **decompress** | string | • **request**<br>• **response** | • **disable**<br>• **enable** |

| Target | Type | Valid Events | Action |
|---|---|---|---|
| **forward** | string | • **request** | • **reset**<br>• **select**<br>• **clone-pool**<br>• **member**<br>• **nexthop**<br>• **node**<br>• **pool**<br>• **rateclass**<br>• **snat**<br>• **snatpool**<br>• **vlan**<br>• **vlan-id** |
| **http-cookie** | string | • **request** | • **insert**<br>• **name** (required)<br>• **value** (required)<br>• **remove**<br>• **name** (required) |
| **http-header** | string/number | • **request**<br>• **response** | • **insert**<br>• **name** (required)<br>• **value** (required)<br>• **remove**<br>• **name** (required)<br>• **replace**<br>• **name** (required)<br>• **value** (required) |
| **http-host** | string | • **request** | • **replace**<br>• **value** |
| **http-referer** | string | • **request** | • **insert**<br>• **value** (required)<br>• **remove**<br>• **replace**<br>• **value** |
| **http-reply** | string | • **request**<br>• **response** | • **redirect**<br>• **location** (required) |
| **http-set-cookie** | string/number | • **response** | • **insert**<br>• **name** (required)<br>• **domain** |

| Target | Type | Valid Events | Action |
|---|---|---|---|
| | | | • **path**<br>• **value** (required)<br>• **remove**<br>• **name** (required) |
| **http-uri** | string/number | • **response** | • **replace**<br>• **path**<br>• **query-string**<br>• **value** |
| **log** | string/number | • **request**<br>• **response** | • **write**<br>• **message** (required) |
| **pem** | string/number | • **request**<br>• **response** | • **classify**<br>• **application**<br>• **category**<br>• **defer**<br>• **protocol** |
| **request-adapt** | string/number | • **request**<br>• **response** | • **disable**<br>• **enable** |
| **response-adapt** | string/number | • **request**<br>• **response** | • **disable**<br>• **enable** |
| **server-ssl** | string/number | • **request** | • **disable**<br>• **enable** |
| **tcl** | string/number | • **request**<br>• **response** | • **set-variable**<br>• **name** (required)<br>• **expression** (required) |
| **tcp-nagle** | string/number | • **request** | • **disable**<br>• **enable** |

# About Traffic Classes

## About traffic classes

The BIG-IP® system includes a feature known as traffic classes. A traffic class is a feature that you can use when implementing optimization profiles for modules such as the Application Acceleration Manager™.

A *traffic class* allows you to classify traffic according to a set of criteria that you define, such as source and destination IP addresses. In creating the traffic class, you define not only classification criteria, but also a classification ID. Once you have defined the traffic class and assigned the class to a virtual server, the system associates the *classification ID* to each traffic flow. In this way, the system can regulate the flow of traffic based on that classification.

When attempting to match traffic flows to a traffic class, the system uses the most specific match possible.

### Creating a traffic class

By creating a traffic class and assigning it to a virtual server, you can classify traffic according to a set of criteria that you specify.

1. On the Main tab, click **Local Traffic** > **Traffic Class**.
2. Click the **Create** button.
3. In the **Name** field, type a name for the traffic class.

   Traffic class names are case-sensitive and can contain letters, numbers, and underscores (_) only.
4. In the **Classification** field, type a text string that the system applies to data flows that match the traffic-class criteria.
   When values from a traffic flow match the criteria specified on this screen, the system tags the traffic flow with this classification value.
5. In the **Source Address** field, type an IP address for the system to match against incoming traffic.
6. For the **Source Mask** field, type a network mask for the specified source address.
7. For the **Source Port** setting, type a port number in the field or select a port name from the list.
   If you select a port name from the list, the system displays the corresponding port number in the text field.
8. In the **Destination Address** field, type an IP address for the system to match against the traffic destination.
9. For the **Destination Mask** field, type a network mask for the specified destination address.
10. For the **Destination Port** setting, type a port number in the field or select a port name from the list.
    If you select a port name from the list, the system displays the corresponding port number in the text field.
11. For the **IP Protocol** setting, type a protocol number in the field or select a protocol name from the list.
    If you select a protocol name from the list, the system displays the corresponding protocol number in the text field.
12. Click the **Finished** button.

After you define the traffic class and assign the class to a virtual server, the system associates the corresponding classification ID to traffic flows that match the specified criteria. In this way, the system can regulate the flow of traffic based on that classification.

# Dynamic Ratio Load Balancing

## Introduction to dynamic ratio load balancing

You can configure Dynamic Ratio load balancing for pools that consist of RealNetworks® RealServer™ servers, Microsoft® Windows® servers equipped with Windows Management Instrumentation (WMI), or any server equipped with an SNMP agent such as the UC Davis SNMP agent or Windows 2000 Server SNMP agent.

To implement Dynamic Ratio load balancing for these types of servers, BIG-IP® Local Traffic Manager™ provides a special monitor plug-in file and a performance monitor for each type of server. The exception is a server equipped with an SNMP agent. In this case, the BIG-IP system provides the monitor only; no special plug-in file is required for a server running an SNMP agent.

### Monitor plug-ins and corresponding monitor templates

This table shows the required monitor plug-in and the corresponding performance monitor types.

| Server Type | Monitor plug-in | Monitor Type |
|---|---|---|
| RealServer™ Windows® server | `F5RealMon.dll` | Real Server |
| RealServer UNIX server | `f5realmon.so` | Real Server |
| Windows server with WMI | `f5isapi.dll` or<br>`F5Isapi64.dll` or<br>`F5.IsHandler.dll` | WMI |
| Windows 2000 Server server | SNMP agent | SNMP DCA and SNMP DCA Base |
| UNIX server | UC Davis SNMP agent | SNMP DCA and SNMP DCA Base |

## Overview of implementing a RealServer monitor

For RealSystem® Server systems, the BIG-IP® system provides a monitor plug-in that gathers the necessary metrics when you have installed the plug-in on the RealSystem Server system. Configuring a RealSystem Server for Dynamic Ratio load balancing consists of four tasks:

- Installing the monitor plug-in on the RealSystem Server system
- Configuring a Real Server monitor on the BIG-IP system
- Associating the monitor with the server to gather the metrics
- Creating or modifying the server pool to use Dynamic Ratio load balancing

### Installing the monitor plug-in on a RealSystem server system (Windows version)

This task installs the monitor plug-in on a RealSystem Server system (Windows version).

1. Download the monitor plug-in `F5RealServerPlugin.dll` from the BIG-IP® system.

   The plug-in is located in the folder `/usr/local/www/docs/agents`.
2. Copy `F5RealServerPlugin.dll` to the RealServer plug-ins directory. (For example, `C:\Program Files\RealServer\plug-ins`.)
3. If the RealSystem Server process is running, restart it.

Once the plug-in is installed and compiled, you must configure a Real Server monitor, associate the configured monitor with the node (a RealSystem Server server), and set the load balancing method to Dynamic Ratio.

### Installing and compiling a Linux or UNIX RealSystem server monitor plug-in

This task installs and compiles a Linux or UNIX RealSystem Server monitor plug-in.

1. Using the **.iso** image, burn a CD-ROM of the BIG-IP® system software.
2. On the CD, navigate to the directory `/downloads/rsplug-ins`.
3. Copy the file `F5RealMon.src.tar.gz` to the directory `/var/tmp` on the BIG-IP system.
4. On the BIG-IP system, change to the directory `/var/tmp`: `cd /var/tmp`
5. Use the UNIX **`tar`** command to uncompress the file `F5RealMon.src.tar.gz`:
   For example, you might enter `tar -xvzf F5RealMon.src.tar`.
6. Change to the **F5RealMon.src** directory: `cd F5RealMon.src`
7. Type the `ls` command to view the directory contents.
8. To compile the source, use the instructions in the file `build_unix_note`.
9. Start RealSystem Server.

Once the plug-in is installed and compiled, you must configure a Real Server monitor, associate the configured monitor with the node (a RealSystem Server server), and set the load balancing method to Dynamic Ratio.

## Overview of implementing a WMI monitor

For Windows running Windows Management Instrumentation (WMI), the BIG-IP® system provides a Data Gathering Agent for the IIS server. Configuring a Windows platform for Dynamic Ratio load balancing consists of these tasks:

- Installing the Data Gathering Agent on the IIS server
- Configuring a WMI monitor on the BIG-IP system
- Associating the monitor with the server to gather the metrics
- Creating or modifying the server pool to use the Dynamic Ratio load balancing method

***Important:*** *To enable a user to access WMI metrics on a Windows server, you must configure the WMI monitor on the BIG-IP system correctly.*

The procedure for installing the Data Gathering Agent on an IIS server differs depending on whether the server is running IIS version 5.0, 6.0, or 7.0, and whether the Data Gathering Agent is the file `f5isapi.dll` (or `f5isapi64.dll`), or the file `F5.IsHandler.dll`.

***Tip:*** *F5 Networks® recommends that you install only the Data Gathering Agent file that pertains to your specific configuration. Installing multiple Data Gathering Agent files could result in unwanted behavior.*

## IIS version support for the data gathering agent files

The procedure for installing the Data Gathering Agent on an IIS server differs depending on whether the server is running IIS version 5.0, 6.0, or 7.0, and whether the Data Gathering Agent is the file `f5isapi.dll` (or `f5isapi64.dll`), or the file `F5.IsHandler.dll`. This table shows each of the Data Gathering Agent files and the IIS versions that support each file.

| Data Gathering Agent | IIS version 5.0 | IIS version 6.0 | IIS version 7.0 |
|---|---|---|---|
| `f5isapi.dll` (32-bit)<br>`f5isapi64.dll` (64-bit) | Yes | Yes | N/A |
| `F5.IsHandler.dll` (32-bit, 64-bit, and .NET) | N/A | Yes | Yes |

## Installing the Data Gathering Agent f5Isapi.dll or f5isapi64.dll on an IIS 5.0 server

You can install the file `f5isapi.dll` or `f5isapi64.dll` on IIS versions 5.0 or 6.0.

***Important:*** *Do not install either of these files on IIS version 7.0 or 7.5. For IIS servers running version 7.0 or 7.5, install the file `F5.IsHandler.dll` instead.*

Perform this task to install the Data Gathering Agent `f5Isapi.dll` or `f5isapi64.dll` on an IIS 5.0 server.

1. Download the **Data Gathering Agent** (`f5Isapi.dll` or `f5isapi64.dll`) from the BIG-IP® system to the Windows platform.

   You can find this plug-in in either the `/var/windlls` or the `/usr/local/www/docs/agents` directory on the BIG-IP system.
2. Copy `f5isapi.dll` or `f5isapi64.dll` to the directory `C:\Inetpub\scripts.`
3. Open the Internet Services Manager.
4. In the left pane of the Internet Services Manager, open the folder `machine_name\Default Web Site\Script`, where *`machine_name`* is the name of the server you are configuring.
   The contents of Scripts folder opens in the right pane.
5. In the right pane, right-click **f5isapi.dll** or **f5isapi64.dll**, and select **Properties**.
   The Properties dialog box for `f5isapi.dll` or `f5isapi64.dll` opens.
6. Clear **Logvisits**. (Logging of each visit to the agent quickly fills up the log files.)
7. Click the File Security tab.
   The File Security options appears.
8. In the **Anonymous access and authentication control group** box, click **Edit**.
   The Authentication Methods dialog box opens.
9. In the dialog box, clear all check boxes, then select **Basic Authentication**.
10. In the Authentication Methods dialog box, click **OK** to accept the changes.

11. In the Properties dialog box, click **Apply**.
    The WMI Data Gathering Agent is now ready to be used.

Once you have installed the plug-in, you must configure a WMI monitor, associate the configured monitor with the pool member, and set the load balancing method to Dynamic Ratio.

## Installing the Data Gathering Agent f5isapi.dll or f5isapi64.dll on an IIS 6.0 server

You can install the file `f5isapi.dll` or `f5isapi64.dll` on IIS versions 5.0 or 6.0.

***Important:*** *Do not install either of these files on IIS version 7.0 or 7.5. For IIS servers running version 7.0 or 7.5, install the file `F5.IsHandler.dll` instead.*

Perform this task to install the Data Gathering Agent `f5isapi.dll` or `f5isapi64.dll` on an IIS 6.0 server.

1. Create a `scripts` directory under the web site document root (`C:\InetPub\wwwroot` for *`Default Website`*).
2. Set the properties of the `scripts` directory to **scripts and executables**.
3. Copy the file `f5isapi.dll` or `f5isapi64.dll` to the created scripts directory.
4. Start IIS manager (**inetmgr**) and navigate to the `scripts` directory.
5. On the right pane, select the file name `f5isapi.dll` or `f5isapi64.dll`.
6. Select **Properties** > **File Security** > **Authentication and Access Control** and ensure that the settings **anonymous user** and **Basic Authentication** are selected.
7. If you want to allow all unknown extensions, then in IIS Manager, navigate to **Web Server Extensions** > **All Unknown ISAPI extensions** and allow all unknown extensions. Otherwise, proceed to step 8.
8. If you want to allow the file `f5isapi.dll` or `f5isapi64.dll` only, navigate to **Web Server Extensions** > **Tasks: Add a New Webserver Extension**. Then:
   a) For the **Name** setting, select **F5 ISAPI** and click **Add** for the required files. This requests a path to the file.
   b) Browse to the file `f5isapi.dll` or `f5isapi64.dll`, using the path `C:\InetPub\wwwroot\scripts\f5isapi.dll` for Default Website, and click **OK**.
   c) Select the **Set Extension Status to Allowed** check box, and click **OK**.
      The value **F5 ISAPI** should now appear in the extensions list as **Allowed**.

Once you have installed the plug-in, you must configure a WMI monitor, associate the configured monitor with the pool member, and set the load balancing method to Dynamic Ratio.

## Installing the Data Gathering Agent F5.IsHandler.dll on an IIS 6.0 server

This task installs the Data Gathering Agent `F5.IsHandler.dll` on an IIS 6.0 server.

1. Create a `scripts`directory under the directory `C:\Inetpub`. (`C:\Inetpub\scripts`).
2. Create a `\bin` directory under the `scripts` directory (`C:\Inetpub\scripts\bin`).
3. Set the properties of the `scripts` directory to **scripts and executables**.
4. Copy the file `F5.IsHandler.dll` to the directory `C:\Inetpub\scripts\bin`.

   In the `C:\Inetpub\scripts` directory, create the file `web.config`. This example shows a `web.config` file on an IIS server running version 6.0. In the example, the path value `f5isapi.dll`, although appearing to be incorrect, is actually correct. It is the type value, `F5.IsHandler`, that directs the server to the correct file.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.web>
        <httpHandlers>
            <clear />
            <add verb="*" path="f5isapi.dll" type="F5.IsHandler" />
        </httpHandlers>
    </system.web>
</configuration>
```

5. From the Start menu, choose Control Panel and double-click **Administration Tools**.
6. Double-click **Internet Information Services**.
   This opens the IIS Management Console.
7. Expand the name of the local computer.
8. Allow the file `ASP.NET v2.0`*`build_number`*:
   a) Select **Web Server Extensions**.
   b) Select **ASP.NET v2.0*build_number***.
   c) Click **Allow**.

9. Create a new virtual directory named `scripts`:
   a) Expand **Websites** and *`Default Web Site`*.
   b) Right-click *`Default Web Site`*, choose **New**, and choose **Virtual Directory**.
   c) Click **Next**.
   d) Type `scripts` for the alias and click **Next**.
   e) Type the name of the directory you created in step 1 (`C:\Inetpub\scripts\`) and click **Next**.
   f) Click **Next** again.
   g) Click **Finished**.

10. Create an application pool for the file `F5.IsHandler.dll`:
    a) Right-click **Application Pools**, select **New**, and select **Application Pool**.
    b) Type `F5 Application Pool` in the **Application Pool ID** field and click **OK**.

11. Right click **scripts** and select **properties**.
12. Set up the application pool:
    a) Click **Virtual Directory**.
    b) From the **Application Pool** list, select **F5 Application Pool**.

13. Set up the mappings:
    a) Click the **Configuration** button.
    b) On the Mappings tab of the Application Configuration screen, click **Add**.
    c) For the executable, type the file name `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll`.
    d) For the file name extension, type `.dll`.
    e) Clear the check box for **Check that file exists** and click **OK**.
    f) On the Application Configuration screen, click **OK**.

14. Set up directory security:

a) Click **Directory Security**.
b) Click the **Edit** button.
c) Disable authentication by clicking the **Anonymous Access and Integrated Windows** box.
d) Select the **Basic Authentication** check box and click **OK**.

***Note:*** *If you are not authenticating locally, you might need to set the default domain or realm.*

**15.** Set up the ASP.NET program:
a) Click the **ASP.NET** tab.
b) From the ASP.NET version list, select **2.0.*`buildnumber`*** (for example 2.0.50727).

**16.** On the scripts Properties page, click **OK**.

**17.** Set up access to the IIS metabase:
a) Run the command `aspnet_regiis -ga <ASP.NETUsername>`.
b) See the web site `http://support.microsoft.com/?kbid=267904`.

Once you have installed the plug-in, you must configure a WMI monitor, associate the configured monitor with the pool member, and set the load balancing method to Dynamic Ratio.

## Installing the Data Gathering Agent F5.IsHandler.dll on an IIS 7.0 server

***Important:*** *Do not install the files `f5isapi.dll` or `f5isapi64.dll` on IIS version 7.0.*

This task installs the Data Gathering Agent `F5.IsHandler.dll` on an IIS 7.0 server.

1. Create a `scripts` directory under the directory `C:\Inetpub`. (`C:\Inetpub\scripts`).
2. Create a `\bin` directory under the `scripts` directory (`C:\Inetpub\scripts\bin`).
3. Copy the file `F5.IsHandler.dll` to the directory `C:\Inetpub\scripts\bin`.
4. In the `C:\Inetpub\scripts` directory, create the file `web.config`. This figure shows an example of `web.config` file on an IIS server running version 7.0.

```
<?xml version="1.0" encoding="UTF-8"?>
    <configuration>
        <system.webServer>
            <handlers>
                <clear />
                    \<add name="F5IsHandler"
                         path="f5isapi.dll"
                         verb="*"
                         type="F5.IsHandler"
                         modules="ManagedPipelineHandler"
                         scriptProcessor=""
                         resourceType="Unspecified"
                         requireAccess="Script"
                         preCondition="" />
            </handlers>
            <security>
                <authentication>
                    <anonymousAuthentication enabled="false" />
                </authentication>
            </security>
         </system.webServer>
```

```
      </configuration>
```

***Important:*** *In this example, the path value `f5isapi.dll`, although appearing to be incorrect, is actually correct. It is the type value, `F5.IsHandler`, that directs the server to the correct file.*

5. Allow anonymous authentication to be overridden by using the `appcmd` command to set the override mode in the machine-level `applicationHost.config` file.

```
                        appcmd set config "Default Web Site/scripts"
                       /section:anonymousAuthentication /overrideMode:Allow

                        /commit:APPHOST
```

6. Set up a new application pool for the file `F5.IsHandler.dll`:
   a) From the Start menu, choose Control Panel.
   b) Choose Administrative Tools
   c) Choose Internet Information Services (IIS) Manager.
   d) From **Connections**, expand *`MachineName`* (*`MachineName\UserName`*).
   e) Right click the **Application Pools** menu and choose **Add Application Pool**.
   f) In the **Name** field, type `F5 Application Pool`.
   g) Click **OK**.

7. Create a new application named `scripts`:
   a) Expand **Web Sites and *`MachineName`***.
   b) Right-click *`MachineName`* and choose **Add Application**.
   c) In the **Alias** field, type `scripts`.
   d) To change the application pool, click **Select**.
   e) For the physical path, type the directory you created in step 1 (`C:\Inetpub\scripts\`).
   f) Click **OK**.

8. Change the **Authentication** setting to **Basic Authentication**:
   a) Select **scripts**.
   b) In the center pane, double click **Authentication**.
   c) Verify that the status of all items under **Authentication** is **Disabled**, except for the **Basic Authentication** item. To enable or disable an authentication item, right click the name and choose Enable or Disable.

Once you have installed the plug-in, you must configure a WMI monitor, associate the configured monitor with the pool member, and set the load balancing method to Dynamic Ratio.

## Installing the Data Gathering Agent F5.IsHandler.dll on an IIS 7.5 server

***Important:*** *Do not install the files `f5isapi.dll` or `f5isapi64.dll` on IIS version 7.5. For IIS servers running version 7.5, always install the file `F5.IsHandler.dll`.*

This task installs the Data Gathering Agent `F5.IsHandler.dll` on an IIS 7.5 server.

1. Create a `scripts` directory under the directory `C:\Inetpub` (`C:\Inetpub\scripts`).
2. Create a `\bin` directory under the `scripts` directory (`C:\Inetpub\scripts\bin`).
3. Copy the file `F5.IsHandler.dll` to the directory `C:\Inetpub\scripts\bin`.
4. In the `C:\Inetpub\scripts` directory, create the file `web.config`.

```
<?xml version="1.0" encoding="UTF-8"?>
                    <configuration>
                        <system.webServer>
                            <handlers>
                                <clear />
                                <add name="F5IsHandler" path="f5isapi.dll"
verb="*" type="F5.IsHandler"
                                    modules="ManagedPipelineHandler"
scriptProcessor="" resourceType="Unspecified"
                                    requireAccess="Script" preCondition=""
/>
                            </handlers>
                            <security>
                                <authentication>
                                   <anonymousAuthentication enabled="false"
 />
                                </authentication>
                            </security>
                        </system.webServer>
                        </configuration>
```

***Important:*** *In the above example, the* **path** *value `f5isapi.dll`, although appearing to be incorrect, is actually correct. It is the* **type** *value, `F5.IsHandler`, that directs the server to the correct file.*

5. Allow anonymous authentication to be overridden by using the `appcmd` command to set the override mode in the machine-level `applicationHost.config` file.

```
appcmd set config "Default Web Site/scripts"
/section:anonymousAuthentication
/overrideMode:Allow /commit:APPHOST
```

***Note:*** *`appcmd` is located in `\windows\system32\intesrv`.*

6. Set up a new application pool for the file `F5.IsHandler.dll`:
   a) From the **Start** menu, choose **Control Panel**.
   b) Choose **Administrative Tools**.
   c) Choose **Internet Information Services (IIS) Manager**.
   d) From **Connections**, expand *`MachineName`* (*`MachineName\UserName`*).
   e) Right click the **Application Pools** menu and choose **Add Application Pool**.
   f) In the **Name** field, type `F5 Application Pool`.
   g) Click **OK**.
   h) From the **Application Pools** list, right click **F5 Application Pool** and choose **Advanced Settings**.
   i) Under the **Process Model** List, click **Identity**, and then click the button to the right of **ApplicationPoolIdentity**.
   j) From **Built-in account** select **NetworkService**.
   k) Click **OK**.
   l) Click **OK**.

7. Create a new application named `scripts`:

a) Expand **Web Sites and *`MachineName`***.
b) Right click *`MachineName`* and choose **Add Application**.
c) In the **Alias** field, type `scripts`.
d) Change the application pool, click **Select**, select **F5 Application Pool** from the **Application Pool** list, and click **OK**.
e) For the physical path, type the directory you created in step 1 (`C:\Inetpub\scripts\`).
f) Click **OK**.

**8.** Change the **Authentication** setting to **Basic Authentication**:

a) Select **scripts**.
b) In the center pane, double click **Authentication**.
c) Verify that the status of all items under **Authentication** is **Disabled**, except for the **Basic Authentication** item. To enable or disable an authentication item, right click the name and choose **Enable** or **Disable**.

Once you have installed the plug-in, you must configure a WMI monitor, associate the configured monitor with the pool member, and set the load balancing method to Dynamic Ratio.

# Legal Notices and Acknowledgments

## Legal Notices

### Publication Date

This document was published on July 12, 2018.

### Publication Number

MAN-0538-00

### Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

### Trademarks

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, Application Acceleration Manager, Application Security Manager, APM, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Certified [DESIGN], F5 Networks, F5 SalesXchange [DESIGN], F5 Synthesis, f5 Synthesis, F5 Synthesis [DESIGN], F5 TechXchange [DESIGN], Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, iApps, IBR, iCall, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, LineRate, LineRate Systems [DESIGN], Local Traffic Manager, LROS, LTM, Message Security Manager, MobileSafe, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, SalesXchange, ScaleN, SDAC (except in Japan), SDC, Signalling Delivery Controller, Solutions for an application world, Software Designed Applications Services, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, Traffix (except Germany), Traffix [DESIGN] (except Germany), Transparent Data Reduction, UNITY, VAULT, vCMP, VE F5 [DESIGN], Versafe, Versafe [DESIGN], VIPRION, Virtual Clustered Multiprocessing, WebSafe, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

### Patents

This product may be protected by one or more patents indicated at:
*http://www.f5.com/about/guidelines-policies/patents*

### Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

### RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

### FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

### Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

### Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

## Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product includes software developed by Brian Gladman, Worcester, UK Copyright ©1998-2010. All rights reserved. The redistribution and use of this software (with or without changes) is allowed without the payment of fees or royalties provided that:

- source code distributions include the above copyright notice, this list of conditions and the following disclaimer;
- binary distributions include the above copyright notice, this list of conditions and the following disclaimer in their documentation.

This software is provided "as is" with no explicit or implied warranties in respect of its operation, including, but not limited to, correctness and fitness for purpose.

This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. Copyright ©1990-1994 Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY

OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Ian Gulliver ©2006, which is protected under the GNU General Public License, as published by the Free Software Foundation.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

This product includes ec2-tools software, copyright © 2008, Amazon Web Services, and licensed under the Amazon Software License. A copy of the License is located at http://aws.amazon.com/asl/ .

This product includes crypto.js software, copyright © 2009-2013, Jeff Mott, and distributed under the BSD New license.

This product includes jxrlib software, copyright ©2009 Microsoft Corp. All rights reserved. Distributed under the new BSD license.

This product contains OpenLDAP software, which is distributed under the OpenLDAP v2.8 license (BSD3-like).

This product includes Racoon 2 software, copyright © 2003-2005 WIDE Project. All rights reserved. Distributed under a BSD-like license.

This product includes node-uuid software, copyright © 2010-2012, Robert Kieffer, and distributed under the MIT license.

This product includes opencsv software, which is distributed under the Apache 2.0 license.

This product includes owasp-jave-encoder software, copyright © 2014, Jeff Ichnowski, and distributed under the New BSD license.

# Index