Progress®
Kemp® LoadMaster®

# Feature Description Health Checking

Generated on: 16 September 2024

# Copyright

**Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend:** Product Documentation Copyright Notice & Trademarks | Progress

# Table of Contents


## Chapter 1: Introduction . . . . . 4

Document Purpose . . . . . 4

Intended Audience . . . . . 5

## Chapter 2: Health Checking. . . . . 6

Service and Non-Service Based Health Checking . . . . . 6

Health Check Types . . . . . 7

Health Check WUI Options . . . . . 12

Health Check Statuses . . . . . 13

Health Check Options. . . . . 14

Custom TCP Service Health Checks . . . . . 27

Example: Binary HTTP Health Check . . . . . 27

Example: Binary SMTP Health Check with Address Verification . . . . . 30

Examples of Log Messages for Common Health Check Issues . . . . . 31

## Chapter 3: References . . . . . 33

# 1

# Introduction

## Introduction

The LoadMaster utilizes health checks to monitor the availability of the Real Servers. If one of the servers does not respond to a health check within a defined time interval for a defined number of times, the weighting of this server is reduced to zero. This zero weighting has the effect of removing the Real Server from the available Real Servers in the Virtual Service until it can be determined that this Real Server is back online.

**Related Links**


- Document Purpose
- Intended Audience


# Document Purpose

## Document Purpose

This document provides information about health checking in the LoadMaster.

# Intended Audience

## Intended Audience

This document is intended to be read by anyone interested in finding out more about health checking in the LoadMaster.

# 2

# Health Checking

## Health Checking

Refer to the sections below for further details on health checking in the LoadMaster.

**Related Links**

- Service and Non-Service Based Health Checking
- Health Check Types
- Health Check WUI Options
- Custom TCP Service Health Checks
- Examples of Log Messages for Common Health Check Issues

# Service and Non-Service Based Health Checking

## Service and Non-Service Based Health Checking

Layer 3 health checks utilize ICMP-based echo requests (pings) to test whether a Real Server can be reached over the network. A Layer 3 check is not Virtual Service specific. For example - when it fails the corresponding Real Server may be removed from all Virtual Services that use it due to a routing issue. If ping fails due to routing – no higher-level health check should pass until the routing issue is resolved.

In contrast to the Layer 3 health checks, service based health checking for both the Layer 4 and Layer 7

health checks are Virtual Service based. When a Real Server fails such a check, it is removed only from the corresponding Virtual Service – all other Virtual Services that use this Real Server are unaffected, as long as their health check passes.

# Health Check Types

## Health Check Types

A number of different health check types are available. With the service health checks, the Real Servers are checked for the availability of the selected service. With TCP/UDP the check is simply a connect attempt.

Refer to the table below for a description of each. The default port for each health check type is listed in parenthesis, but users can specify any valid port to perform the health check on.

| Type | Description |
|---|---|
| ICMP | The LoadMaster sends ICMP echo requests (pings) to the Real Servers. A Real Server fails this check when it does not respond with an ICMP echo response in the configured response time for the configured number of retries. |
| TCP Connection Only | The LoadMaster attempts to open TCP-connection to the Real Server on the configured service port: it sends a TCP SYN packet to the server on the service port. The server passes the check if it responds with a TCP SYN ACK in the response time interval. In this case, the LoadMaster closes the connection by FIN, ACK. If the server fails to respond within the configured response time for the configured number of times, the server is assumed to be down.<br><br>When using **TCP Connection Only** as the **Real Server Check Method**, the health check port will default to the port of the Virtual Service. |
| FTP | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 21). If the server responds with a greeting message with status code 220, the LoadMaster sends a QUIT command to the server, closes the connection and marks it as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different status code, it is assumed down. |

| Type | Description |
|---|---|
| TELNET | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 23). If the server responds with a command string beginning with the char '0xff', the LoadMaster closes the connection and marks the server as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different command string, it is assumed down. For further information on the **Check Parameters**, refer to the Check Parameters section. |
| SMTP | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 25). If the server responds with a greeting message with status code 220, the LoadMaster sends a QUIT command to the server, closes the connection and marks it as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different status code, it is assumed down. |
| HTTP | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 80).<br><br>The LoadMaster sends a HTTP/1.0 HEAD request the server, requesting the page "/".<br><br>If the server sends a HTTP response with a status code of 200-299, 301, 302 or 401 the LoadMaster closes the connection and marks the server as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different status code, it is assumed down.<br><br>It is also possible to specify additional health check status codes which is considered up. This means that if the Real Server responds with one of these codes, the LoadMaster will see the Real Server service as up and can send traffic to it. This can be done using the **Status Code** field. For further information, please refer to the HTTP or HTTPS Protocol Health Checking section.<br><br>HTTP 1.0 and 1.1 support is available. URL health checking is also available. HTTP 1.1 allows you to check host header enabled web servers. |

| Type | Description |
|---|---|
| HTTPS | The LoadMaster opens an SSL connection to the Real Server on the Service port (port 443). The LoadMaster sends a HTTP/1.0 HEAD request the server, requesting the page "/". If the server sends a HTTP response with a status code of 200-299, 301, 302 or 401 the LoadMaster closes the connection and marks the server as active. If the server fails to respond within the configured response time for the configured number of times or if it responds with a different status code, it is assumed down.<br><br>HTTP 1.0 and 1.1 support is available. HTTP 1.1 allows you to check host header enabled web servers. |
| POP3 | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 110). If the server responds with a greeting message that starts with +OK, the LoadMaster sends a QUIT command to the server, closes the connection and marks it as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different status code, it is assumed down. |
| NNTP | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 119). If the server responds with a greeting message with status code 200 or 201, the LoadMaster sends a QUIT command to the server, closes the connection and marks it as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different status code, it is assumed down. |
| IMAP | The LoadMaster opens a TCP connection to the Real Server on the Service port (port 143). If the server responds with a greeting message that starts with "+ OK" or "* OK", the LoadMaster sends a LOGOUT command to the server, closes the connection and marks it as active. If the server fails to respond within the configured response time for the configured number of times, or if it responds with a different status code, it is assumed down. |
| DNS | The **Name Server (DNS) Protocol** value is only available in the **Real Server Check Method** drop-down list when |

| Type | Description |
|---|---|
| | the Virtual Service **Protocol** is set to **udp**. The LoadMaster performs nslookups against an A record on the server over UDP port 53. If the server successfully responds to the DNS query, the LoadMaster marks it as active. If the server fails to respond within the configured response time for the configured number of times or if it responds unsuccessfully to the A record request, it is assumed down. |
| RDP | By default, the remote terminal health check will use the Virtual Service port. The default ports for RDP are 3389 and 3391. RDP health checks start the same way as **TCP Connection Only** to confirm that the port is opened. After this, the LoadMaster checks if it can log in to the Real Server. No username or password is sent – it just checks to see if the service is listening. If a response is received, the Real Server is marked as up. If no response is received the Real Server is marked as down. |
| Binary | Specify a hexadecimal string to send to the Real Server. Specify a hexadecimal string which is searched for in the response sent back from the Real Server. If the LoadMaster finds the pattern in the response, the Real Server is considered up. Specify the number of bytes to search for the reply pattern within.<br><br>There is no default port for this, as it is not a protocol-based check. |
| LDAP | Select an LDAP endpoint to use for the health check. The LDAP health check uses the LDAP credentials and protocol specified in the LDAP endpoint. The health check is run against the Real Server IP address and port. The LDAP health check comprises of a LoadMaster connecting to a Real Server and validating the specified user credentials. The health check is performed in two steps:<br><br>**Step 1: Check if the Real Server specified port is up and available.**<br><br>**Step 2: Attempt to log in to the Real Server using the LDAP specified credentials.** |

| Type | Description |
|---|---|
| | If step 1 and step 2 are true, the health check passes. If step 1 or step 2 fails, the health check fails. |
| None | No health checking is performed. |

Health checks are always seen from the interface address (the active LoadMaster if in High Availability (HA) mode, not the shared IP address). Production traffic could be seen from the client IP address, the Virtual Service address, or the interface address depending on the scenario. For further details, refer to the Routing Feature Description.

The LoadMaster uses health checks that can be specified in the WUI. By default, the best available health check is selected for the Virtual Service, based on the **Service Type**.

| Service | Port | Protocol | Layer |
|---|---|---|---|
| FTP | 21 | TCP | Layer 4/Layer 7 |
| TELNET | 23 | TCP | Layer 4/Layer 7 |
| SMTP | 25 | TCP | Layer 4/Layer 7 |
| HTTP | 80 | TCP | Layer 4/Layer 7 |
| HTTPS | 443 | TCP | Layer 4/Layer 7 |
| POP3 | 110 | TCP | Layer 4/Layer 7 |
| NNTP | 119 | TCP | Layer 4/Layer 7 |
| IMAP | 143 | TCP | Layer 4/Layer 7 |

| Service | Port | Protocol | Layer |
|---|---|---|---|
| DNS | 53 | UDP | Layer 4/Layer 7 |
| ICMP | N/A | TCP | Layer 3 |
| TCP | N/A | TCP | Layer 4 |
| RDP | 3389 | TCP | Layer 7 |
| LDAP | 389/636 | UDP/TCP | Layer 4/Layer 7 |

The **Service Type** selected may limit the available **Real Server Check Method** options. For example, the service type **Remote Terminal** will permit checking with **Remote Terminal Protocol** but this check method is not available for other service types.

For other ports, the LoadMaster uses Layer 4 health checks for TCP services and Layer 3 health checks for **ICMP Ping** for both TCP and UDP Virtual Services. The settings for the health checks can be changed from the default settings using the Virtual Service modify screen to accommodate non-standard settings. For example, one could run a HTTP service on port 8080 instead of 80, and change the health check to HTTP instead of the default Layer 4 check.

# Health Check WUI Options

## Health Check WUI Options

Health checking is enabled by default when a Virtual Service is created. For UDP Virtual Services, ICMP health checking is the default. The other available choices are DNS and none (there is no **Service Type** field for UDP Virtual Services.

For TCP Virtual Services, the default **Service Type** is **Generic** and the default health check is **TCP Connection Only**. The available health checks depend on the selected **Service Type**.

**Related Links**

- Health Check Statuses
- Health Check Options

# Health Check Statuses

# Health Check Statuses




The status of the Virtual Services which exist in the LoadMaster is displayed in the **Virtual Services > View/ Modify Services** screen.

The Virtual Service status may be one of the following:

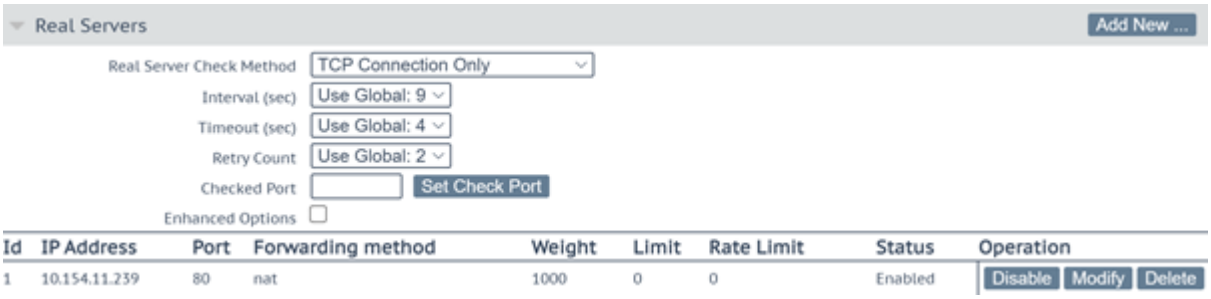
| Status | Description |
|---|---|
| Up | At least one Real Server is available. |
| Down | No Real Servers are available. |
| Sorry | All Real Servers are down and traffic is routed to a separately configured Sorry Server that is not part of the Real Server set and is not health checked. For further details on Sorry Servers, refer to the WUI Configuration Guide. |
| Disabled | The Virtual Service has been administratively disabled by unticking the **Activate or Deactivate Service** check box in the **Basic Properties** section of the Virtual Service modify screen. |
| Redirect | A fixed redirect response has been configured. One way that redirect Virtual Services can be created is by using the **Add a Port 80 Redirector VS** option in the **Advanced Properties** section. |
| FailMsg | An error message has been configured. This status is usually seen while configuring the Edge Security Pack (ESP). An error message can be specified using the **Not** |

| Status | Description |
|---|---|
| | **Available Redirection Handling** options. |
| Unchecked | Health checking of the Real Servers has been disabled. All Real Servers are used. |

**Note:** Health checks are always seen from the interface address (the active LoadMaster if in High Availability (HA) mode, not the shared IP address).For local Real Servers, the LoadMaster uses the interface local to the Real Server. For non-local Real Servers the LoadMaster uses routing schema/logic.

# Health Check Options

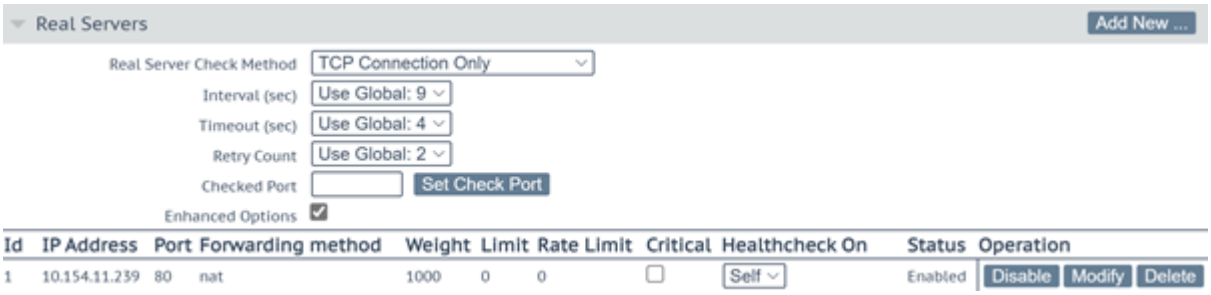
Health check options are set in the Real Servers section in the Virtual Service modify screen in the LoadMaster Web User Interface (WUI) (**Virtual Services > View/Modify Services > Modify**).




This section lists the Real Servers that are assigned to the Virtual Service. The properties of the Real Servers are summarized and there is also the opportunity to add or delete a Real Server, or modify the properties of a Real Server.

### Real Server Check Method

This provides a list of health checks for well-known services, as well as lower level checks for TCP/UDP or ICMP. With the service health checks, the Real Servers are checked for the availability of the selected service. With TCP/UDP, the check is simply a connect attempt.




When one of the **HTTP/HTTPS**, **Generic** or **STARTTLS protocols** Service Types are selected, the following

health check options are available:

- ICMP Ping
- HTTP Protocol
- HTTPS Protocol
- TCP Connection Only
- Mail (SMTP) Protocol
- Network News (NNTP) Protocol
- File Transfer (FTP) Protocol
- Telnet Protocol
- Mailbox (POP3) Protocol
- Mailbox (IMAP) Protocol
- Binary Data
- LDAP
- None

When the **Remote Terminal** Service Type is selected the following health check options are available:

- ICMP Ping
- TCP Connection Only
- Remote Terminal Protocol
- None

---

**Note:** For a UDP Virtual Service, only the **ICMP Ping** and **Name Service (DNS) Protocol** options are available for use.

---

**Check Parameters**

In LoadMaster firmware version 7.2.52, the check **Interval**, **Timeout**, and **Retry Count** settings can be configured on each Virtual Service or SubVS. Previously, these were just global settings. You can configure the global settings in **Rules & Checking > Check Parameters**. The global settings are used by default for all Virtual Services.

**Interval (seconds):** This field specify the number of seconds that will pass between consecutive health check attempts and defines the length of the 'health check cycle'. To override the global interval, you can select the any other value from the drop-down list. The global option cannot be selected if either **Timeout** or **Retry Count** parameters have been set to a value other than the global setting.

**Timeout (seconds):** This is the allowed maximum wait time for a connection to be established and for the receipt of a reply to a health check. If no reply is received before the timeout expires, then the connection is re-attempted as long as the Retry Count has not been reached. If any HTTP response is received in response to the health check, then no retries are attempted in the current health check cycle. To override the global timeout value, you can select the any other value from the drop-down list.

**Retry Count:** This specifies the number of times the health check will be re-attempted if the timeout above is reached before receiving a response. To override the global retry count value, you can select the any other value from the drop-down list. The retry count does not apply if any response is received in response to the

health check. If, for example, either a 200 or a 404 response is received, then no retries are attempted in the current cycle.

To configure these settings for a specific Virtual Service, expand the **Real Servers** section of the Virtual Service or SubVS modify screen. A **Real Server Check Method** must be selected to see the relevant fields. You can configure these settings to either use the global value, set a specific value within the provided range, or reset to the default value.

If you configure these settings for a parent Virtual Service and then create a SubVS within that Virtual Service, the check values are reset to use the global values.

Checked Port

The port to be health checked. If no port is specified, the port of the Real Server is used.

---

**Note:** **Extra Ports** specified in the Virtual Service are not used for Real Server health checking.

---

**Enhanced Options**

Enabling the **Enhanced Options** check box provides additional health check options:

- Minimum number of RS required for VS to be considered up
- Critical
- Healthcheck On

If the **Enhanced Options** check box is disabled (the default), the Virtual Service is considered available if at least one Real Server is available. If the **Enhanced Options** check box is enabled, you can specify the minimum number of Real Servers that must be available to consider the Virtual Service to be available.

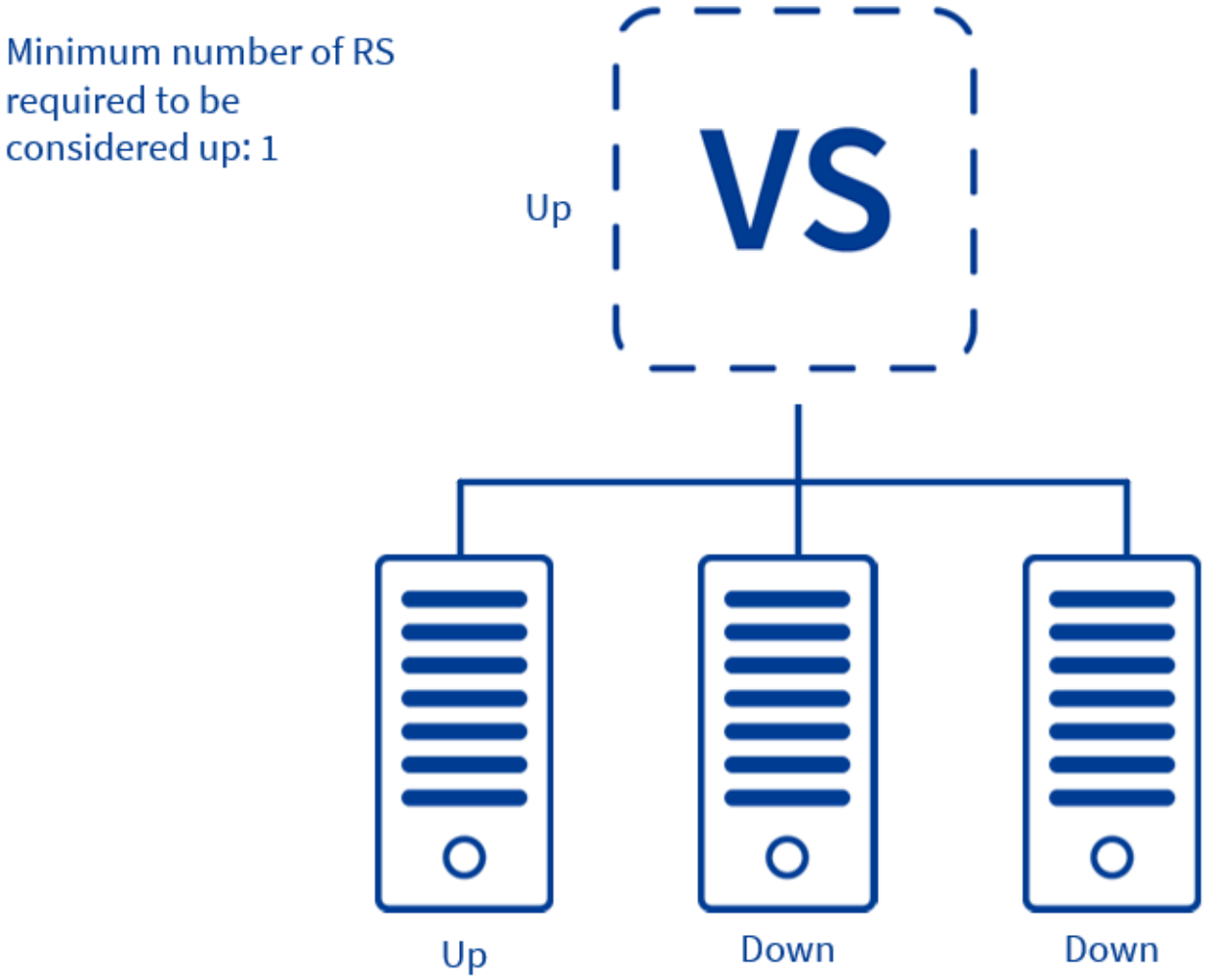
**Minimum number of RS required for VS to be considered up**

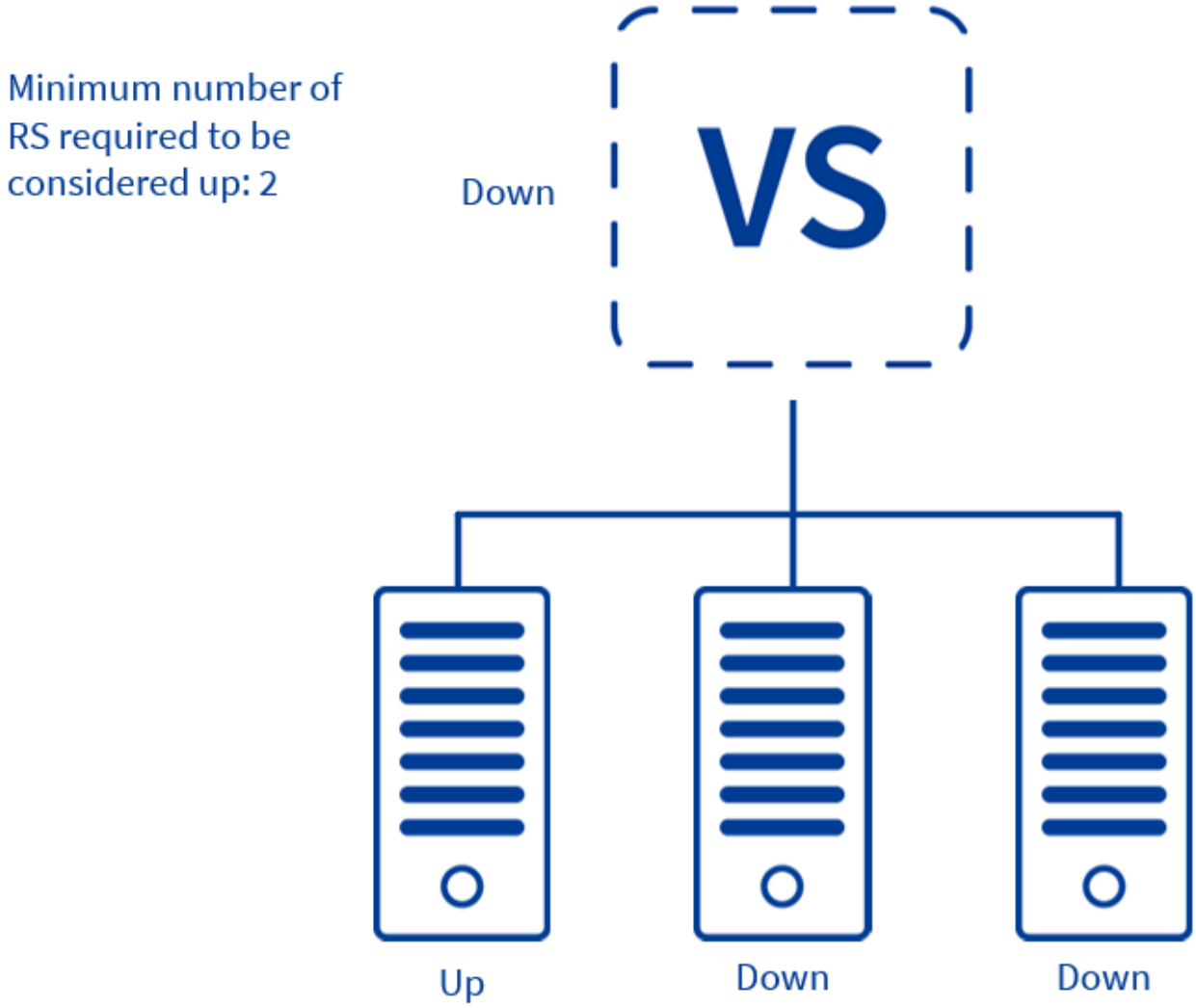
---

**Note:** This option will only appear if the **Enhanced Options** check box is enabled and if there is more than one Real Server.

---

Select the minimum number of Real Servers required to be available for the Virtual Service to be considered up.


Minimum number of RS required to be considered up: 1
Up
VS
Up
Down
Down




If less than the minimum number of Real Servers is available, a critical log is generated and the Virtual Service is marked as down. If some Real Servers are down but it has not reached the minimum amount specified, a warning is logged. If the email options are configured, an email is sent to the relevant recipients. For further information on the email options, refer to the Web User Interface (WUI), Configuration Guide.

---

**Note:** Note that the **Critical** option has a higher priority than the **Minimum number of RS required for VS to be considered up** option. The system marks a Virtual Service as down whenever a Real Server that is marked as **Critical** (that is, the **Critical** check box is enabled) becomes unavailable – even if **Enhanced Options** are enabled and there are more than the specified minimum number of Real Servers still available.

---

In all cases, if the Virtual Service is considered to be down and the Virtual Service has a sorry server or an error message configured, these are used. The sorry server is never health checked, it is assumed to be up.

If the minimum number is set to the total number of Real Servers and one of the Real Servers is deleted, the minimum will automatically reduce by one.

A SubVS is said to be available and can be routed to if and only if the number of available Real Servers

within that SubVS is greater than or equal to the limit.

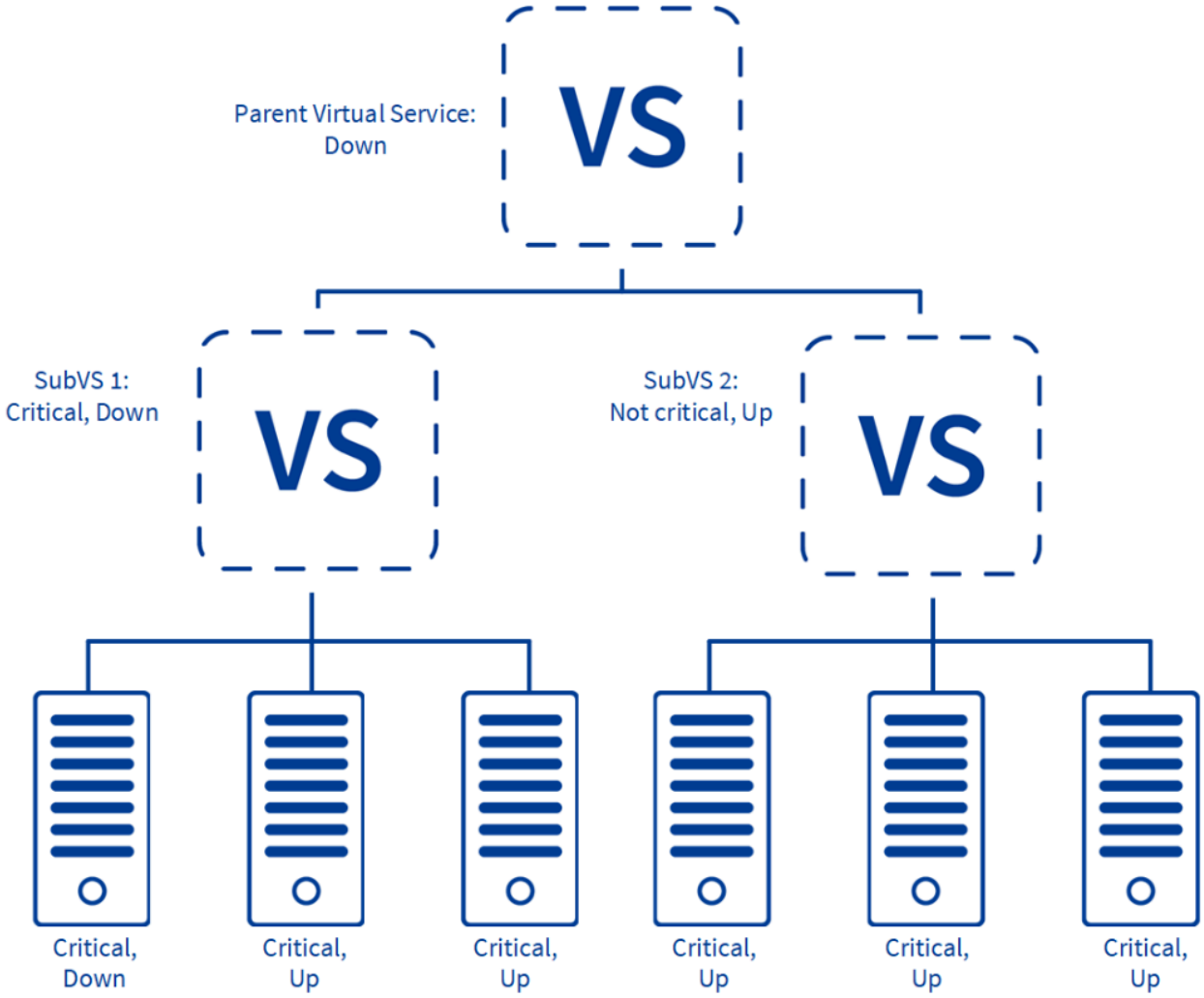
If the number of available Real Servers is below this limit, the SubVS will not accept traffic as it is marked as down for failing the health check and this is logged appropriately.

**Critical**

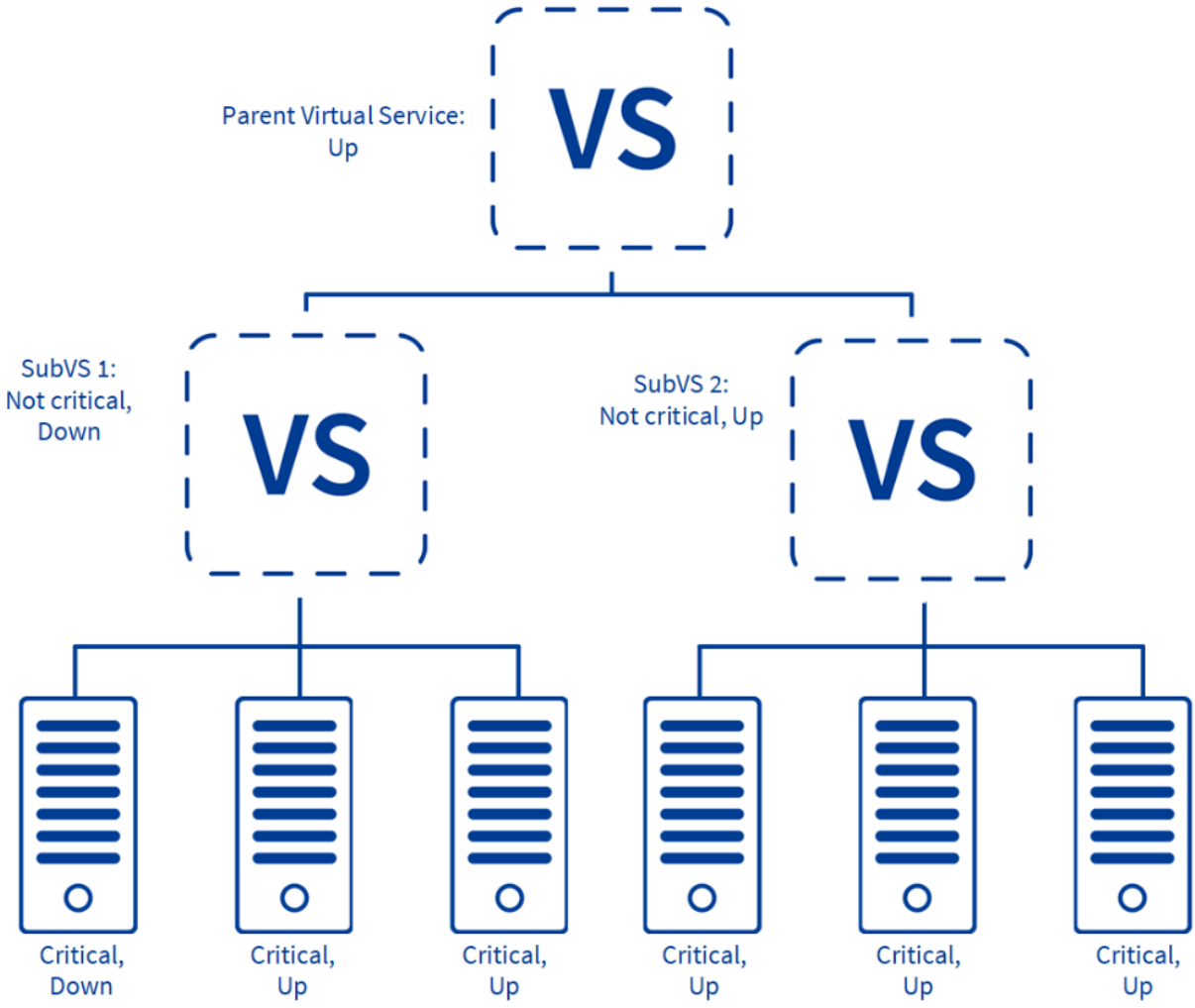
---

**Note:** This option will only appear if the **Enhanced Options** check box is enabled.

---




In the Real Servers section of the Virtual Service modify screen, there is a **Critical** check box for each of the Real Servers. Enabling this option indicates that the Real Server is required for the Virtual Service to be considered available. The Virtual Service is marked as down if the Real Server has failed or is disabled. Therefore, in the example above the parent Virtual Service is down because there is a critical SubVS down. The critical SubVS is down because a critical Real Server is down.




If a Real Server on a SubVS is marked as critical – the SubVS is marked as down if that Real Server is down. However, the parent Virtual Service will not be marked down unless that SubVS is marked as critical.

---

**Note:** The **Critical** option has a higher priority than the **Minimum number of RS required for VS to be considered up** option. For example, if three Real Servers are added and if the minimum is set to two and only one Real Server is down but that Real Server is set to critical – the Virtual Service is marked as down.

---

In all cases, if the Virtual Service is considered to be down and the Virtual Service has a sorry server or an error message configured, these are used.

---

**Note:** It is not possible to perform complex, chained health checking such as AND or OR statements. For example – if there are five Real Servers, it is not possible to say; IF **Real Server 2** is up OR **Real Server 4** is up, mark the Virtual Service as up.

---

**Note:** You can mark both Real Server 2 and Real Server 4 as critical, but if either of them go down, the parent Virtual Service is marked as down.

---

**Healthcheck On**

**Note:** This option will only appear if the **Enhanced Options** check box is enabled.

**Note:** This option is only available on LoadMasters with firmware version 7.1.35 and above.

In the Real Servers section of the Virtual Service modify screen, there is a **Healthcheck On** drop-down list for each of the Real Servers. This allows you to specify what Real Server the health check is based on. This can either be set to **Self** in order to perform the health check based on that particular Real Server status, or another Real Server can be selected. For example – if Real Server 1 is down, any Real Servers which have their health check based on Real Server 1 will also be marked as down, regardless of their actual Real Server status.

In the text below, the term “follow” is used to indicate a health check relationship between two Real Servers. For example – if there are two Real Servers called RS1 and RS2. RS1 is said to “follow” RS2 if RS2 is selected in the **Healthcheck On** drop-down list for the RS1 health check.

Some points to be aware of are listed below:

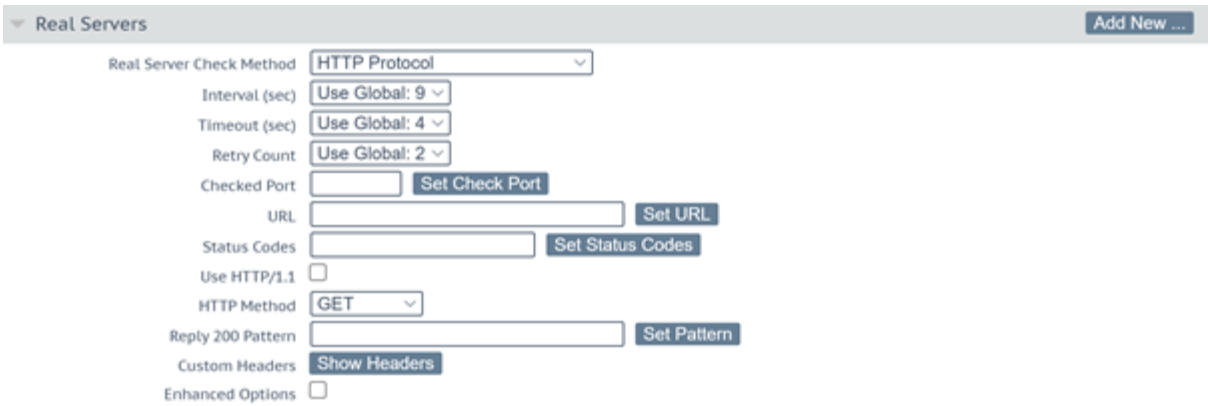
- A Real Server can only follow a Real Server and not a SubVS.
- A Real Server can follow a Real Server that is also following a third Real Server. The status of the first two Real Servers will reflect the status of the third Real Server.
- Chains of Real Servers are allowed – but loops are not allowed and cannot be created.
- If a Real Server is deleted (either singly or as part of a Virtual Service), all Real Servers that are following the Real Server are reset to normal behavior (that is, they will start using the Virtual Service health check options).
- If all Real Servers in a Virtual Service are following Real Servers on a different Virtual Service, the health check parameters for the Virtual Service are not shown on the WUI (because the settings do not affect any Real Servers).
- Disabling the **Enhanced Options** check box will disable all Real Server health check following for that Virtual Service and regular checking on the individual Real Servers will resume. If Session Management is enabled, an entry is recorded in the audit log any time the **Enhanced Options** check box is enabled or disabled.

**Related Links**

- HTTP or HTTPS Protocol Health Checking
- Binary Data Health Checking
- Check Parameters

## HTTP or HTTPS Protocol Health Checking

When either the **HTTP Protocol** or **HTTPS Protocol** options are selected a number of extra options are available as described below.




**Note:** The **post data** option only appears if the **POST HTTP Method** is selected.

**Note:** The **Reply 200 Pattern** option only appears if either the **POST** or **GET HTTP Method** is selected.

**URL**

By default, the health checker tries to access the URL to determine if the machine is available. A different path can be specified here, for example **/owa/healthcheck.htm**.

**Status Codes**

Health check status codes can be set to override default functionality. Without any **Status Codes** set, the following HTTP status codes are considered to be Up:

- 200-299: Success (for example 200 OK, 202 Accepted)
- 301: Moved Permanently
- 302: Found
- 401: Unauthorized

Additionally, 2xx status codes are subject to pattern matching the response data, if this is configured. Other codes are considered up without pattern matching, even if it is set.

If custom health check codes are set:

- Check codes may be set to a list of numbers, each from 300-599
- Check codes may be up to 127 characters long, which means 32 valid codes
- Any code in the list is considered to have a health check status of Up
- Configured codes override the default set

- 
  - 2xx codes are always considered up in all cases and are subject to pattern matching, if configured
  - Check codes may be official HTTP status codes, unofficial codes or custom-defined user codes – as long as they fall in the range of 300-599
  - 
    - For a summary of official HTTP status codes, refer to: http://www.ietf.org/assignments/http-status-codes/http-status-codes.xmlFurther information can be found in the more detailed article:https://tools.ietf.org/rfc/rfc7231.txt

- If the application being checked uses decimal sub-codes (for example 400.1), the decimal part is ignored for the purposes of matching the configured codes. So, 400.1, 400.2 and so on, will all match 400.

- For a list of Microsoft sub-codes using decimals, refer to: https://support.microsoft.com/en-us/kb/943891
- Sub-codes may not be configured in the **Status Codes** field – please use the three digit code
- Sub-codes are grouped by the top-level code

**Use HTTP/1.1**

By default, the LoadMaster uses **HTTP/1.0**. However, you may opt to use **HTTP/1.1**, which will operate more efficiently. When using **HTTP/1.1**, the health checks are multiplexed to a single connection. This means that more health checks are sent to the server in a single connection, which is more efficient from a connection point of view, that is, there is only one connection rather than multiple connections every time the Real Servers are checked.

---

**Note:** Optimization only works on HTTP (not HTTPS) connections, that means no connection multiplexing happens with HTTP/1.1 for HTTPS health checks.

---

When a Real Server does not send any response to a health check request when using the HTTP/1.1 head method, the LoadMaster takes the Real Server out of rotation in all Virtual Services where the Real Server is participating. If this is not the desired behavior, use HTTP/1.0 instead (you could always send a host an extra header if needed).

**HTTP/1.1 Host**

---

**Note:** This field will only be visible if **Use HTTP/1.1** is selected.

---

When using **HTTP/1.1** checking, the Real Servers require a hostname to be supplied in each request. If no value is set, then this value is the IP address of the Virtual Service.

To send Server Name Indication (SNI) host information in HTTPS health checks, please enable **Use HTTP/1.1** in the **Real Servers** section of the relevant Virtual Service(s) and specify a host header. If this is not set, the IP address of the Real Server is used.

**HTTP Method**

When accessing the health check URL, the system can use either the **HEAD**, **GET** or **POST** method.

In LoadMaster firmware version 7.2.52, support was added for the **OPTIONS** method for HTTP and HTTPS health check types. This specifies that the server will be marked up when the LoadMaster receives a 200 OK in response to an HTTP (or HTTPS) OPTIONS request sent by the LoadMaster.

The OPTIONS HTTP method requests a description of the permitted communication options from the server. A 200 OK response from the server contains a response body which can be optionally searched for specific text in order to provide an additional check. To search the response body, specify the search text in the **Reply 200 Pattern** text box that appears when you select the **OPTIONS** HTTP method. The server will be marked up if the provided text is found in the response body; otherwise, the server is marked down.

**Post Data**

This field will only be available if the **HTTP Method** is set to **POST**. When using the **POST** method, up to 2047 characters of POST data can be passed to the server.

**Reply 200 Pattern**

When using the **GET** or the **POST** method, the contents of the returned response message can be checked. If the response contains the string specified by this Regular Expression, then the machine is determined to be up. The response will have all HTML formatting information removed before the match is performed. Only the first 4K of response data can be matched.

The LoadMaster will only check for this phrase if the reply from the server is a 200 code. If the reply is something else, the page is marked as down without checking for the phrase. However, if the reply is a redirect (code 302), the page is not marked as down. This is because the LoadMaster assumes that the phrase will not be present and also it cannot take the service down, as the redirect would then become useless.

Both Regular Expressions and Perl Compatible Regular Expressions (PCRE) can be used to specify strings.

The checker only checks the response one line at a time (and it ignores HTML/XML tags). It is not possible to concatenate multiple lines on the same reply 200 pattern.

For further information on Regular Expressions and PCRE, please refer to the Content Rules, Feature Description document.

**Custom Headers**

Here you can specify up to 4 additional headers/fields which is sent with each health check request. Clicking the **Show Headers** button will show the entry fields. The first field is where you define the key for the custom header that is to be part of the health check request. The second field is the value of the custom header that is to be sent as part of the health check request. Once the information is input, click the **Set Header** button. Each of the headers can be up to a maximum of 20 characters long and the fields can be up to a maximum of 100 characters long. However, the maximum allowed number of characters in total for the 4 header/fields is 256.

The following special characters are allowed in the **Custom Headers** fields:

**; . ( ) / + = - _**

If a user has specified **HTTP/1.1**, the Host field is sent as before to the Real Server. This can be overridden by specifying a Host entry in the additional headers section. The User-Agent can also be overridden in the same manner. If a Real Server is using adaptive scheduling, the additional headers which are specified in the health check are also sent when getting the adaptive information.

**Health Check Authorization**

It is possible to perform a health check using an authenticated user: enable **Use HTTP/1.1**, select **HEAD** as the **HTTP Method** and enter the Authorization header with the correctly constructed value. The Authorization field is constructed as follows:

**The username and password are combined into a string “username:password”.**

**The resulting string is then encoded using the RFC2045-MIME variant of Base64, except not limited to 76 char/line.**

**The authorization method and a space, i.e. “Basic“ is then put before the encoded string.**

For example, if the user agent uses 'Aladdin' as the username and 'open sesame' as the password then the field is formed as follows:

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

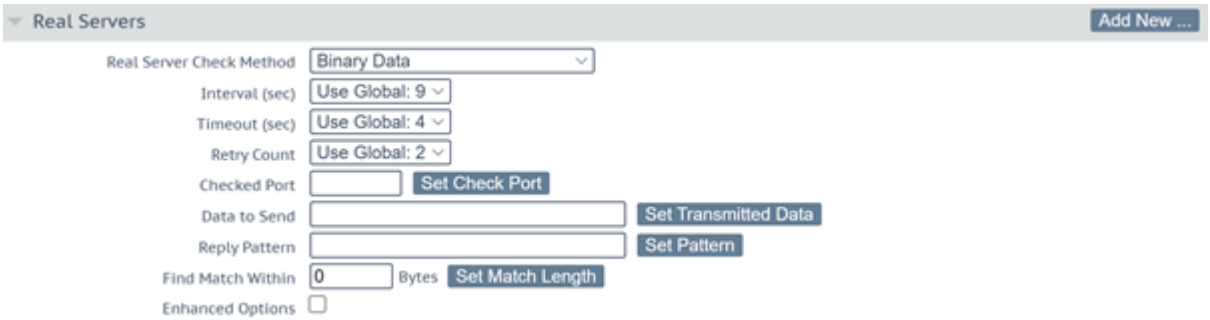
### Using SNI with Health Checks

To send SNI host information in HTTPS health checks, please enable **Use HTTP/1.1** in the **Real Servers** section of the relevant Virtual Service(s) and specify a host header. If this is not set, the IP address of the Real Server is used.

## Binary Data Health Checking

When **Binary Data** is selected as the health check method, some other fields are available, as described below.

For examples on how to create custom, protocol-specific, binary health checks, refer to the Custom TCP Service Health Checks section.




### Data to Send

Specify a hexadecimal string to send to the Real Server. This hexadecimal string must contain an even number of characters. For example:

- **12d** is not valid because it has an odd number of characters
- **12de** is valid and will result in 16bits of data being sent to the Real Server
- **012d** is valid and results in 16 bits of data being sent to the Real Server
- **00012d** is valid and results in 24 bits of information

### Reply Pattern

Specify the hexadecimal string that the LoadMaster will attempt to locate in the response data. The response is treated as a sequence of bytes; there is no header or body – there is just a sequence of bytes. This checker could be used to health check an Extended Binary Coded Decimal Interchange Code (EBCDIC) data stream.

If the LoadMaster finds the pattern in the response, the Real Server is considered up. If the string is not found, the Real Server is marked as down.

---

**Note:** This hexadecimal string must contain an even number of characters.

---

**Find Match Within**

When a response is returned, the LoadMaster will search for the **Reply Pattern** in the response. The LoadMaster will search up to the number of bytes specified in this field for a match.

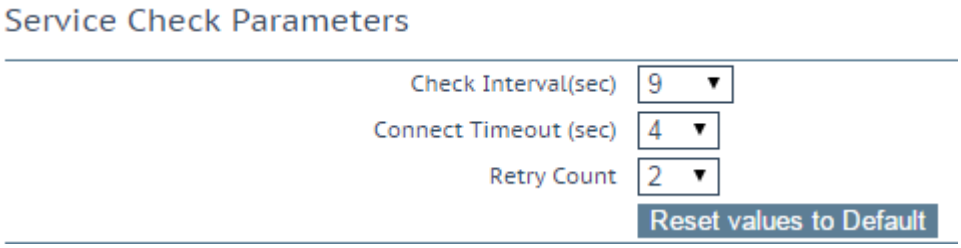
Setting this to **0** means that the search is not limited. Data is read from the Real Server until a match is found. The first 8 KB is read from the Real Server.

Setting the value to more than the length of the reply string means that the check will act as if the value has been set to **0**, i.e. all packets (up to 8 KB) are searched.

## Check Parameters

To access the **Check Parameters** screen, go to **Rules & Checking > Check Parameters** in the main menu of the LoadMaster WUI. These parameters are global and apply across all health checks on all Virtual Services.

The LoadMaster utilizes Layer 3, Layer 4 and Layer 7 health checks to monitor the availability of the Real Servers and the Virtual Services.




**Check Interval(seconds)**

This field specifies the number of seconds that will pass between consecutive health check attempts and defines the length of the 'health check cycle'.

The recommended value is **9** seconds. The minimum value for the **Check Interval** is **Connect Timeout** * **Retry Count** + 1, i.e. (4*2) + 1 = 9.

**Connect Timeout (seconds)**

This is the allowed maximum wait time for a connection to be established and for the receipt of a reply to a health check. If no reply is received before the timeout expires, then the connection is re-attempted as long as the Retry Count has not been reached. If any HTTP response is received in response to the health check, then no retries are attempted in the current health check cycle. The recommended value is **4** seconds. If spurious timeouts are observed for health checks, this number may need to increase.

**Retry Count**

This specifies the number of times the health check will be re-attempted if the timeout above is reached before receiving a response. To override the global retry count value, you can select any other value from the drop-down list. The retry count does not apply if any response is received in response to the health check. If, for example, either a 200 or a 404 response is received, then no retries are attempted in the current cycle.

In LoadMaster firmware version 7.2.52, the check **Interval**, **Timeout**, and **Retry Count** settings can be configured on each Virtual Service. Previously, these were just global settings.

# Custom TCP Service Health Checks

## Custom TCP Service Health Checks

The LoadMaster provides a robust set of pre-defined health checks for many TCP-based applications as described elsewhere in this document. When you have an application in your infrastructure for which there is no pre-defined health check, or if you want to perform content verification for any protocol, a binary health check can be created that performs this task.

For example, you have a custom application in your configuration and that application provides health status on a particular port on the server on which it is running. It may return something as simple as an integer or it may return something more complex, like several thousand bytes of text. In either case, you can construct a binary health check that will:

**Connect to the server on the appropriate port for the status program.**

**Send a string to the server (if required) to trigger the return of the status information.**

**Search the server response for a specified string that indicates that the service is up.**

**Mark the server up or down, as appropriate.**

**Related Links**

- Example: Binary HTTP Health Check
- Example: Binary SMTP Health Check with Address Verification

# Example: Binary HTTP Health Check

## Example: Binary HTTP Health Check

This section provides an example of constructing a basic version of the pre-defined HTTP health check using a binary health check. This example intends to describe how a binary health check is configured and how it works. It illustrates how the mechanism works and how you can use it to construct an arbitrary TCP service health check.

The HTTP health check essentially issues an HTTP 'GET' command and then makes an up/down decision for the server based on the server response code and text. You can construct a similar binary health check (minus the response code checking) that follows the flow below:

**Attempt to open a TCP connection to the server IP:port.**

**The HTTP server should respond and complete the connection; if it does not, it is marked down.**

**If the connection is successful, LoadMaster sends a 'GET /index.html' command to the server.**

**If the server responds, our health check searches for the text 'It works!' in the first 1024 bytes of the response. If this text is found, the server is marked up; otherwise, the server is marked down.**

Of course, step 2 above implies that you know how the server responds when the service running on the server is available. If you do not have prior knowledge of how it will respond, It is usually possible to discover this through:

- Command line interaction with the server.
- Watching server traffic using a traffic analyzer.

For example, in the case of interacting with an HTTP server as in this example, you can use the **telnet** command to connect to the server, issue the same GET command as above, and watch what the server returns. The following shows an example of such an exchange:

**$ telnet 192.168.150.250 80GET /index.html**

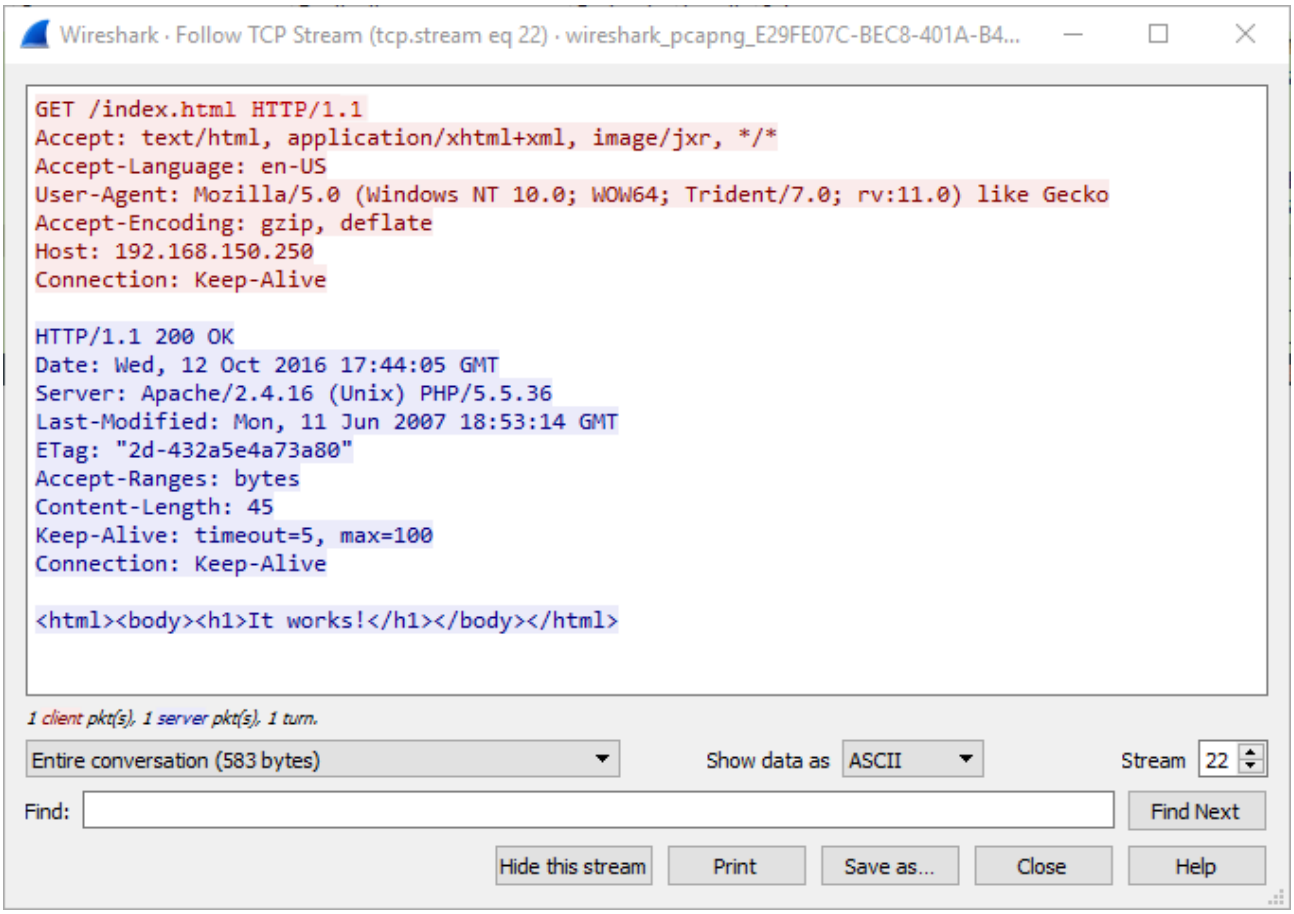
**<html><body><h1>It works!</h1></body></html>**

**Connection to host lost.**

You type the **telnet** command; when the connection is made to the server, the screen is reset (usually with character echo turned off). You then type '**GET /index.html<Enter>**'. You must press the **Enter** key to send the command to the server. The server then sends its response (a small HTML page) and drops the connection.

Not all applications support command line connections and in these cases you can examine client-server traffic in detail by taking a network packet capture using a tool like Wireshark. Continuing with our example above, you could:

- **Run Wireshark to capture network traffic on a system that is connected to the same networks as the client and the server, or on the server itself.**
- **Open a browser on a client system to http://192.168.150.250 and wait for the page to display.**
- **Stop the packet capture on Wireshark and look for the first packet that has the IP address of the system running the browser you used above as the source IP and the server as the destination IP (in our example, 192.168.150.250). Then, right-click that packet and select Follow > TCP Stream from the drop-down menu that appears. This should open a screen in Wireshark that looks something like this:**




The two callouts in the screenshot above highlight the pieces of the conversation between the client (at top) and the server (at bottom) that you need for the binary health check. However, you do not want the ASCII text -- you must convert that to hexadecimal for input into the LoadMaster WUI.

- **You can do this using a few methods:**

- Using the **Show data as** drop-down in the screen above
- Using an ASCII to HEX converter (many free converters are available on the web)
- Using a simple ASCII chart (again, freely available on the web)

For the **Data to Send** in the binary health check UI example, specify the HEX equivalent of **GET /index.html** followed by a carriage return and line feed characters (required by an HTTP server to terminate the command line). The table below shows the hexadecimal string at the top and the ASCII string at the bottom:

| **47** | **45** | **54** | **20** | **2f** | **69** | **6e** | **64** | **65** | **78** | **2e** | **68** | **74** | **6d** | **6c** | **0d** | **0a** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | E | T | | / | I | n | d | e | x | . | h | t | m | l | <CR> | <LF> |

Note that each ASCII character is specified by two hexadecimal digits (for example, the letter G is

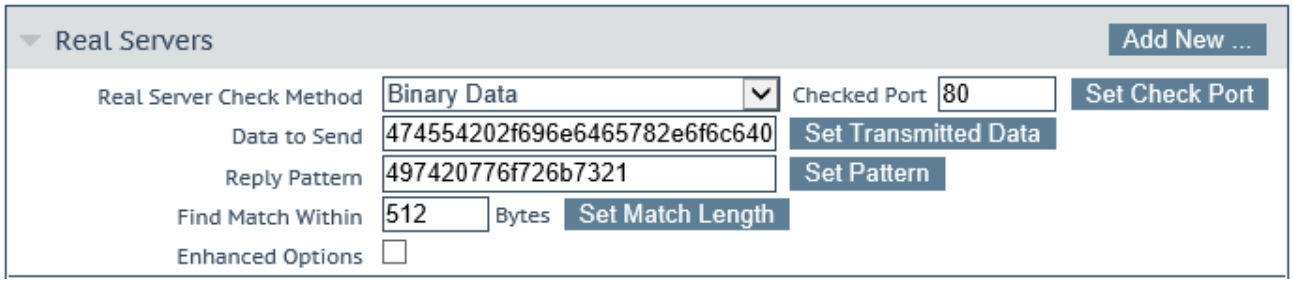
represented by a hexadecimal '47' in the above). No spaces are permitted when you add this to the UI, so the following string is what you copy into the **Data to Send** text box:

**474554202f696e6465782e68746d6c0d0a**

Similarly, the hexadecimal **Reply Pattern** to specify for **It works!** would be the following:

**497420776f726b7321**

The screenshot below shows how a binary health check created using the above parameters looks in the UI.




# Example: Binary SMTP Health Check with Address Verification

# Example: Binary SMTP Health Check with Address Verification

The pre-defined SMTP health check on LoadMaster works by opening a TCP connection to an SMTP server; it sets the up/down status of the server depending on whether the connection is successful or not.

In this section, you will create a custom health check that connects to the server and checks basic SMTP server operations by requesting verification of a specific user name from the server. The binary health check will:

**Connect to the SMTP server.**

- **Send an SMTP command (VRFY) along with a specific mail user name (for example, 'postmaster'). The server, if it is working properly, should respond positively to the verification request.**
- **Mark the server as up (if the server responds and a positive response is found to the query above); otherwise, it will be marked down.**

First we will look at how the above exchange appears on the wire; again, as in the previous example, either telnet or Wireshark will do. In telnet, the exchange looks like this:

**$ telnet 192.168.150.250 25220 Methuselah.local ESMTP PostfixVRFY postmaster252 2.0.0 postmasterQUIT221 2.0.0 ByeConnection to host lost.**

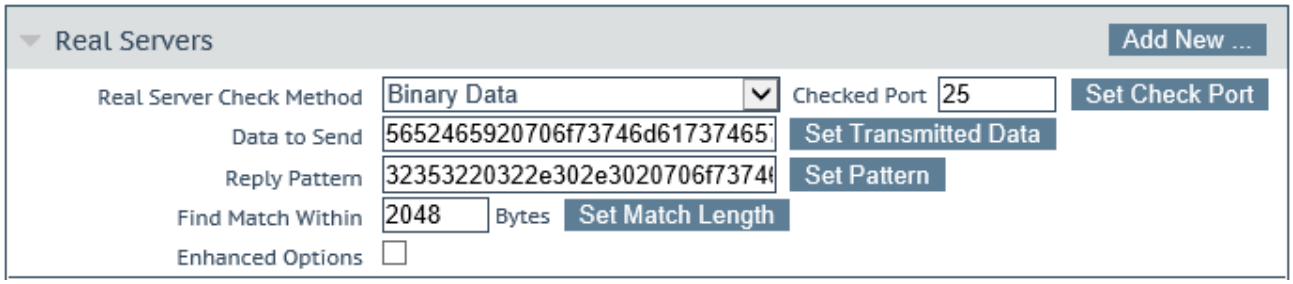
VRFY postmaster is the **Data to Send** to the SMTP server, and the next line shown above is the expected **Reply Pattern** when the server is working properly.

To configure this into the LoadMaster WUI, you need the hexadecimal equivalent of the above:

**Data to Send: 5652465920706f73746d61737465720d0a**

**Reply Pattern: 32353220322e302e3020706f73746d6173746572**

The following screenshot shows how the binary health check settings should appear after configuring the settings above into the WUI:




# Examples of Log Messages for Common Health Check Issues

## Examples of Log Messages for Common Health Check Issues

The log format is as follows:

```
<Date> <Time> <LoadMaster host name> Removing RS <Real server IP:port> from VS
<VS name> - <reason>
```

Examples of log messages for common health check issues and explanations for these logs are provided below.

**Connection Failed - unreachable host**

```
2023-03-01T12:05::00+00:00 lb100 l4d: Removing RS 192.168.10.20:443 from VS
192.168.10.10:443(HTTPS) - Connection Failed - unreachable host
```

There is no route to the host.

**Timeout waiting for connection**

```
2023-03-01T12:05::00+00:00 lb100 l4d: Removing RS 192.168.10.20:443 from VS
192.168.10.10:443(HTTPS) - Timeout waiting for connection
```

Three-way handshake (SYN, SYN-ACK, ACK) did not complete successfully.

**Timeout waiting for data**

```
2023-03-01T12:05::00+00:00 lb100 l4d: Removing RS 192.168.10.20:443 from VS
192.168.10.10:443(HTTPS) - Timeout waiting for data
```

Three-way handshake (SYN, SYN-ACK, ACK) completed successfully, a request was sent to the Real

Server but the LoadMaster did not receive a reply.

# 3

# References

## References

Unless otherwise specified, the following documents can be found at https://docs.progress.com/.

**Web User Interface (WUI), Configuration Guide**

**Content Rules, Feature Description**

**Port Following, Feature Description**