



Microsoft Terminal Services & Coyote Point Systems Equalizer

Prepared by: Mark Hoffmann
Coyote Point Systems Inc.



Table of Contents

INTRODUCTION	3
BENEFITS	3
PREREQUISITES / COMPONENTS	3
DEPLOYMENT GUIDELINE EXAMPLE	4
CONFIGURATION EXAMPLE 1: MSTS/EQUALIZER DEPLOYMENT WITHOUT SESSION SERVICES	5
CONFIGURATION EXAMPLE 2: MSTS/EQUALIZER DEPLOYMENT WITH IP REDIRECT SESSION SERVICES	12
CONFIGURATION EXAMPLE 3: MSTS/EQUALIZER DEPLOYMENT WITH TOKEN REDIRECT SESSION SERVICES	15
FOR ADDITIONAL HELP	19
APPENDIX A: TERMINAL SERVICES SERVER AGENT	20

Copyright © 2009-2010 Coyote Point Systems, Inc.

All Rights Reserved.

Equalizer is a trademark of Coyote Point Systems Incorporated. All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION AND ANY ASSOCIATED PRODUCTS.

Introduction

This is a general deployment guideline for load balancing Microsoft Terminal Services with Coyote Point Systems Equalizer™. This deployment guideline assumes the reader is familiar with the general functionality of both Equalizer™ and Microsoft Windows Server 2003 and 2008.

This guideline is broken into sections that describe how to configure Equalizer and Microsoft Terminal Server 2003 and 2008 (MSTS) in the following configurations:

- MSTS 2000, 2003, and 2008 server farms that do not use Session Directory/Session Broker
- MSTS 2003 and 2008 server farms that use IP-based session redirection with Session Directory/Session Broker
- MSTS 2008 server farms that use token-based session redirection with Session Directory or Session Broker

Benefits

- Improve MSTS Server or application availability and uptime
 - Reduce single points of failure for your MSTS servers or applications
 - Automated recovery from MSTS Server failures
- Improve MSTS Server performance
 - Load balancing ensures more consistent server and application response
 - Improve hardware utilization by sending new MSTS application requests to the servers with the fewest connections in the application farm
- Improve deployment flexibility
 - No single subnet requirement for MSTS deployments as required by Microsoft Network Load Balancer

Prerequisites / Components

- Microsoft Terminal Services running on Windows Server Standard, Advanced or Enterprise 2003 or 2008
- Coyote Point Systems Equalizer™ model E250si or greater, software version 7.2.4d or greater. The steps defined in this guideline reference an Equalizer running version 8.5.0b of software. If you are configuring a different version of Equalizer and you are unfamiliar with the corresponding steps for your version, please reference the Equalizer administration guide available in the **Help** menu of your Equalizer management interface.

Deployment Guideline example

All the examples in this deployment guideline use the same basic network topology as shown in Figure 1.1, and behave in the following manner:

- Terminal server or remote desktop clients connect to a virtual cluster presented by the Equalizer.
- Connections from clients *without* an existing MSTS "session" are load balanced by Equalizer, according to the policy selected, across a farm of MSTS servers.
- Connections from clients *with* an existing MSTS "session" are connected to the same MSTS server previously used, allowing the client to resume an existing MSTS session containing saved system and application states.

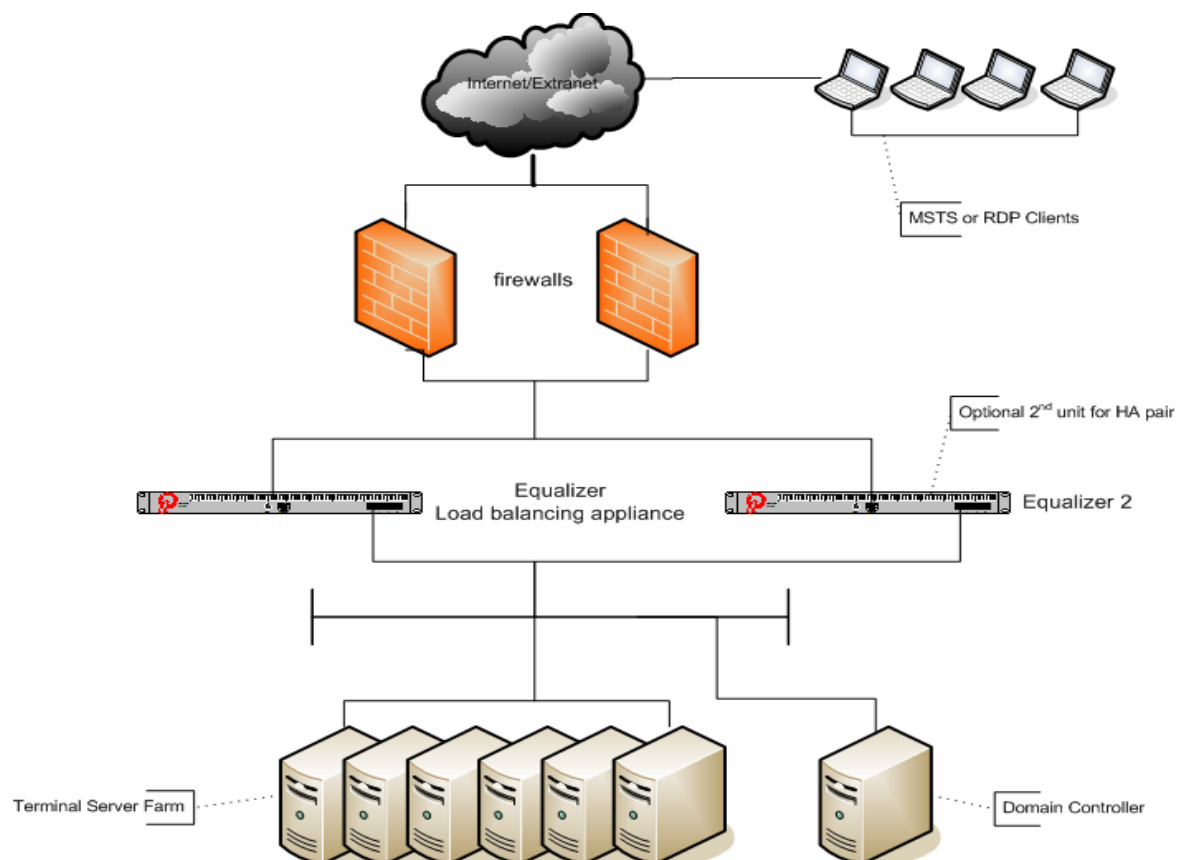


Figure 1.1 - Sample dual network connection diagram

Configuration Example 1: MSTS/Equalizer Deployment without Session Services

Overview

In an MSTS server farm deployment with no Session Directory or Session Broker services, users log into a group of MSTS servers that have the same applications installed, and their files are usually kept on administrator-defined shared storage. Users may leave a session behind that they want to resume later, but there is no Session Directory (MSTS 2003) or Session Broker (MSTS 2008) server configured.

To place Equalizer in front of such a group of independent MSTS servers, a Layer 4 cluster on Equalizer is configured with a **sticky time** that is equal to the session time set on the MSTS servers. The first time a client connects to the MSTS cluster, it is load balanced according to the policy set on the cluster. If the client disconnects and leaves a session behind on the MSTS server, it will be resumed the next time the client connects before the session (and sticky) timers expire.

For example, say ClientA connects to the cluster and is load balanced to ServerA. The client disconnects and leaves a session behind. The default expiration time for a session on MSTS is "Never". If you leave the session timer set to "Never", you'll probably want to set the sticky timer on the cluster to a suitably large value (like, say, a month or year), so that the client can resume the session they left behind whenever they get around to it. If ServerA goes down, the session is lost; Equalizer detects the server is down and routes the connection to another server, ServerB.

Leaving the MSTS session timer set to 'Never' and the sticky timer to a large value means, however, that the client's connection is now going to stick to ServerB until the server goes down or Equalizer reboots. To ensure that Equalizer's load balancing distributes clients efficiently among the MSTS servers behind it, the MSTS session timer can be set to a value between 1 minute and 5 days, and a corresponding value set for the cluster's sticky timer. This way, client connections are re-load balanced at a specified interval.

Also, the **max connections** value on each server defined on Equalizer should be set to the maximum connections setting on the associated real MSTS server *minus one*, unless server agents are used. See Step 5, below.

Deployment Steps

The following steps are required to set up the example configuration involve defining the Equalizer cluster and the associated parameters, and defining the servers and server settings:

Equalizer Cluster

1. Define the Terminal Services virtual cluster
2. Set **sticky time** for Terminal Services cluster
3. Setting the LB algorithm for the MSTS Cluster

Cluster Servers

4. Add real Terminal Servers to virtual cluster
5. Set **max connections** value for each Terminal Server

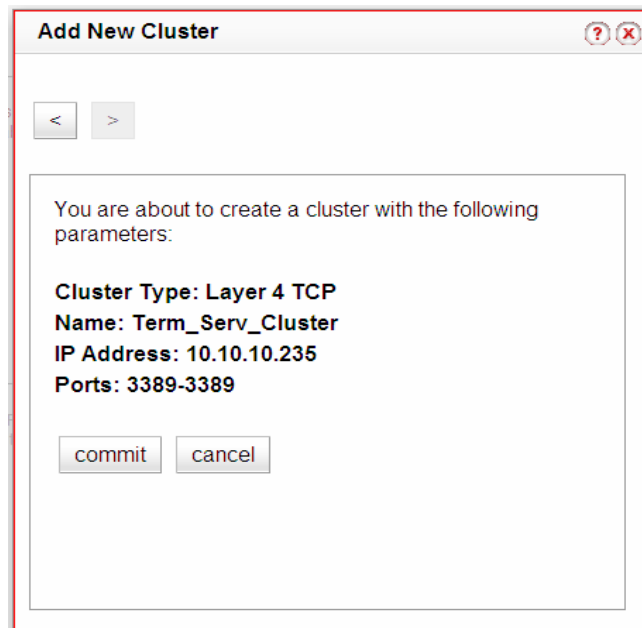
MS Terminal Servers

6. Set the **session time** on each Terminal Server to the same value used for **sticky time** on the Equalizer cluster.
7. Set the **maximum connections** value on each Terminal Server to the same value used for **max connections** on the corresponding server definition on Equalizer.

Step 1: Defining the Terminal Services virtual cluster

Begin by defining the virtual cluster on Equalizer.

1. Open the Equalizer Administrative Interface by pointing a browser at one of the IP addresses assigned to Equalizer's network interfaces. Log in using the default **touch** login and password.
2. In the upper left pane of the interface, right click on the **Equalizer** icon and select **Add Cluster** from the drop down menu. The screen shown below appears.



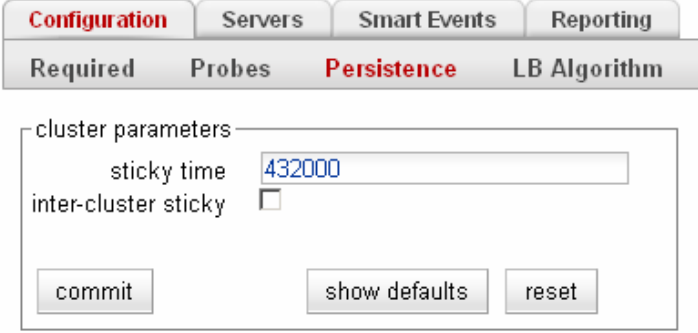
[For help with this (or any) Equalizer wizard screen, Click ? > Context Help from the menu at the top right of the wizard popup.]

3. Select **Layer 4 TCP** and click the **Next [>]** icon.
4. Name the cluster and assign an IP consistent with the single or dual network deployment model you chose for your configuration.
5. Finally, set the **start port** for the cluster to 3389 (leave **end port** blank). This is the default TCP port used by RDP clients to connect to MSTS.

Step 2: Set 'sticky time' for the Terminal Services Cluster

Next, we'll define the **sticky time** for our L4 cluster. This value represents the amount of time a client will be returned to the same MSTS server for the resumption of an existing session. For this example we will use **5 days** expressed in seconds (**432000 seconds**). This is equal to the largest setting below 'Never' that is supported by MSTS (and set in Step 6).

1. In the left pane, click on the newly created Terminal Services cluster.
2. In the right panel click on the **Persistence** tab (see Figure 1.3).
3. Enter **432000** into the **sticky time** text box.
4. Click **commit** to save the change.



Configuration Servers Smart Events Reporting

Required Probes **Persistence** LB Algorithm

cluster parameters

sticky time

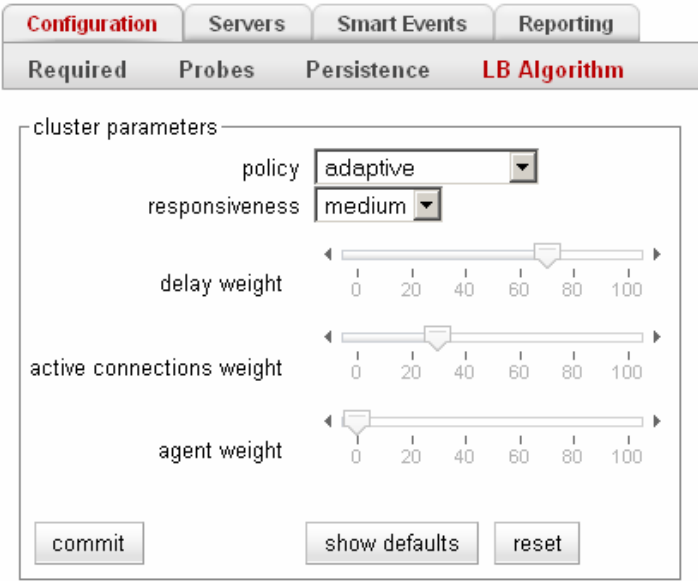
inter-cluster sticky

Figure 1.3 - Set sticky time

Step 3: Set LB algorithm for the Terminal Services Cluster

Next, we'll define the LB algorithm for our L4 cluster. For this example we will use the adaptive algorithm.

1. In the left pane, click on the newly created Terminal Services cluster.
2. In the right frame click on the **LB Algorithm** tab - see Figure 1.4.
Select the **policy** and **responsiveness** that you want; the sliders indicate the relative importance of the load balancing parameters listed for the chosen policy. You can also choose a **custom** policy and set the sliders as you like.
[Click Help > Context Help from the menu at the top right of the Equalizer screen for help with the settings on this tab.]
3. Click **commit** to save your changes.



Configuration Servers Smart Events Reporting

Required Probes Persistence **LB Algorithm**

cluster parameters

policy

responsiveness

delay weight

active connections weight

agent weight

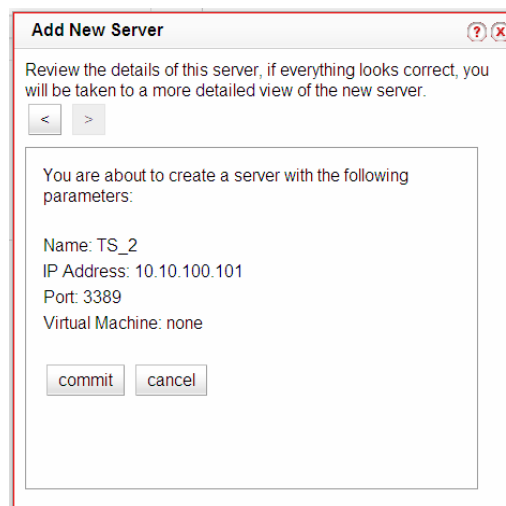
Figure 1.4 - LB algorithm

Deployment Guide

Note: The **adaptive** policy evaluates server response time and the active connection count and dynamically adjusts the traffic flowing to the server accordingly.

Step 4: Add servers to the Terminal Services Cluster

1. In the upper left pane, right click on the newly created Terminal Services cluster and select 'Add Server' from the popup menu - see Figure 1.5
2. In the fields of the 'Add New Server' wizard enter a server **name** and a server **IP address** that corresponds with the IP address of an MSTS server.
3. Set the server **port** to **3389** then click the **Next** [>] icon in the wizard.
4. Review the details for this server and if all values are correct click **commit**.
5. Repeat the above steps until you have added all of your MSTS servers to the cluster.



Step 5: Set Max Connections

The value to use for **max connections** on Equalizer is determined by the capacity of the real server and the number of Terminal Server licenses installed for that server. Please refer to Microsoft's [Terminal Server Capacity and Scaling](#) documentation and your hardware manufacturer's specifications for calculating the number of connections per server best suited for your MSTS environment. A similar setting is used on the Terminal Server itself (See Step 7, below).

By default, Equalizer uses ICMP and TCP probes to determine if a terminal server is up or down. If the probe port is set to port 3389, Equalizer will make a TCP connection to port 3389 during each probe cycle (default: 10 seconds). This connection to the terminal server will use up one of the server's licensed connections -- and if Equalizer cannot connect because all the licensed terminal services connections are being used, it will mark the server down and drop all existing connections.

Deployment Guide

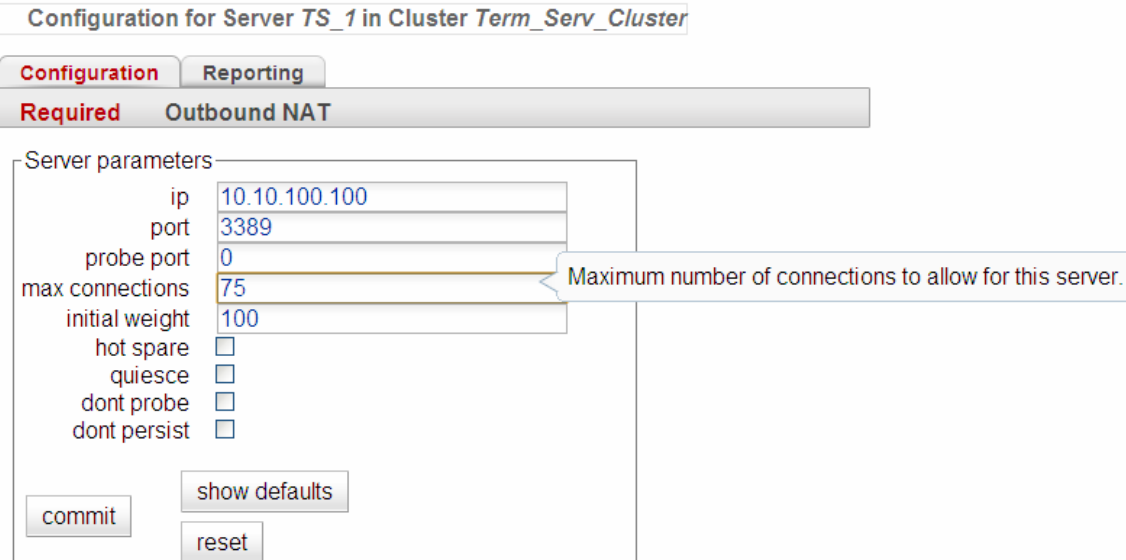
Therefore, when using TCP probes to the Terminal Service port (3389), the maximum connections setting on Equalizer must be set to the maximum connections setting on the terminal server *minus one*.

If you *don't* want to consume a licensed connection when probing a terminal server, you can change the probe port to another port on the server that will open a TCP connection when Equalizer connects to that port. Doing so, however, doesn't mean that Terminal Services is actually running on port 3389.

You can avoid using a Terminal Services license for probing and provide accurate server status to Equalizer by using a "server agent", a small program that runs on the server. See Appendix A for more information.

To set **max connections** for all the servers in a cluster, do the following:

1. In the left pane of the Administrative Interface, click on the plus sign next to the newly created MSTS Cluster to display all the servers.
2. Click on one of the servers to display its configuration tabs (see Figure 1.6).
3. In the right pane enter a **max connections** value for this server. For this example we will be using 75.
Repeat the above process for each server defined in your cluster.



Configuration for Server TS_1 in Cluster Term_Serv_Cluster

Configuration Reporting

Required Outbound NAT

Server parameters

ip	10.10.100.100
port	3389
probe port	0
max connections	75
initial weight	100
hot spare	<input type="checkbox"/>
quiesce	<input type="checkbox"/>
dont probe	<input type="checkbox"/>
dont persist	<input type="checkbox"/>

Maximum number of connections to allow for this server.

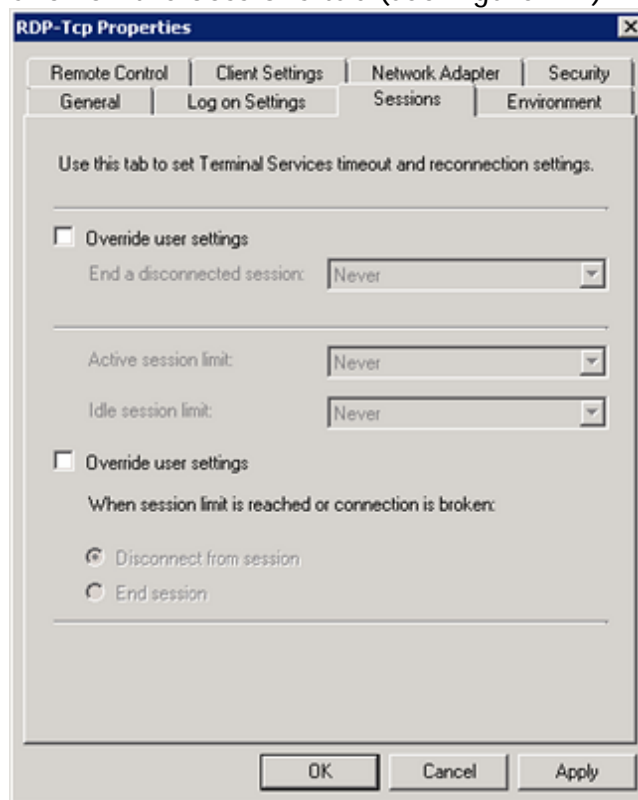
commit show defaults reset

Step 6: Set MS Terminal Server Session Time

1. On each MSTS server, open the Server Manager by clicking Start > Programs > Administrative Tools > Server Manager.

Deployment Guide

2. In the left panel of the **Server Manager**, click the plus sign next to **Roles** to expand the Roles list, then click the plus sign next to **Terminal Services** to expand that list.
3. Click on **Terminal Services Configuration**.
4. In the right panel, under **Connections**, right-click on the Connection Name (default is **RDP-Tcp**) and select **Properties**.
5. Click on the **Sessions** tab (see Figure 1.7):



The screen above shows the default session options.

6. Click on **Override user settings** to enable the check box, and select a session time equal to the sticky time you set in Step 2. Continue with the next step.

Step 7: Set MS Terminal Server Maximum Connections

1. Click on the **Network Adapter** tab. Click on **Maximum connections** and set the same value you set for **max connections** on Equalizer in Step 5 (in our example, 75).
2. Click **OK** to save your settings and return to the **Server Manager**.

Configuration Example 2: MSTS/Equalizer Deployment with IP Redirect Session Services

Overview

The obvious pitfall of the previous example configuration is that the power of Equalizer's load balancing algorithms (a key advantage over using MS load balancing solutions like NLB or Session Broker load balancing) gets applied to a particular client at a specified interval set by the MSTS session timer and cluster sticky timer, and not on each new login.

This disadvantage is overcome by employing MSTS Session Broker (on MSTS 2008; Session Directory on MSTS 2003) to manage sessions across the server farm. Equalizer and MSTS session services cooperate to allow an existing server session to continue on client reconnect, even if Equalizer initially load balances the client connection to another server.

In contrast to the previous configuration, this configuration requires:

- An Equalizer cluster **sticky time** of 0.
- If running on MSTS 2008, each MSTS server has **Session Broker Load Balancing disabled**.
- On both MSTS 2003 and 2008, each MSTS server must have **IP redirection enabled**.
- For session redirection to work, clients must appear to Equalizer and the servers behind it to be on the same subnet as Equalizer's internal interface (in single network mode) or external interface (in dual network mode).

MSTS servers must be able to connect directly to clients on the internal or external interface subnet, depending on the network mode. This may require NAT on firewalls between clients and Equalizer, a VPN, or special routing, depending on your network configuration.

- The **spoof** option must be disabled on the cluster.

Deployment Steps

The steps required to set up the example configuration involve defining the Equalizer cluster and the associated parameters, and defining the servers and server settings:

Deployment Guide

Equalizer Cluster

1. Define the Terminal Services virtual cluster
2. Disable the **spoof** option on the cluster
3. Setting the LB algorithm for the MSTs Cluster

Cluster Servers

4. Add real Terminal Servers to virtual cluster
5. Set max connection value for each Terminal Server

MS Terminal Servers

6. Set the **session time** on each Terminal Server to any value desired, or leave the default (Never).
7. Set the **maximum connections** value on each Terminal Server to the same value used for **max connections** on the corresponding server definition on Equalizer.
8. Configure MS Terminal Server for **IP Redirect** session reconnection.

Step 1: Defining the Terminal Services virtual cluster

Follow the instructions in Step 1 of **Configuration Example 1**.

Step 2: Disable 'spoof' for the Terminal Services Cluster

1. In the left pane, click on the name of the Terminal Services cluster.
2. In the right pane, disable the **spoof** flag by clearing the check box next to it.
3. Click **commit** to save the change.

Step 3: Set LB algorithm for the Terminal Services Cluster

Follow the instructions in Step 3 of **Configuration Example 1**.

Step 4: Add real servers to Terminal Services Cluster

Follow the instructions in Step 4 of **Configuration Example 1**.

Step 5: Set Max Connections

Follow the instructions in Step 5 of **Configuration Example 1**.

Step 6: Set MS Terminal Server Session Time

Do one of the following:

- If you want to leave the default value of 'Never' for the session time, skip this step and proceed with **Step 7**.
- If you want to set the session time to a lower value, follow the instructions in **Step 6** of **Configuration Example 1**.

Step 7: Set MS Terminal Server Max Connections

Follow the instructions in Step 7 of **Configuration Example 1**.

Step 8: Configure MS Terminal Server for IP Redirect Session Reconnection

In this step, we configure each MSTS server to join a Session Directory (MSTS 2003) or Session Broker (MSTS 2008) server farm; this enables session redirection. We'll also configure IP redirection for session reconnects.

For MSTS 2008 Session Broker configurations:

1. On each MSTS server, open the Server Manager by clicking **Start > Programs > Administrative Tools > Server Manager**.
2. In the left panel of the Server Manager, click the plus sign next to **Roles** to expand the Roles list, then click the plus sign next to **Terminal Services** to expand that list.
3. Click on **Terminal Services Configuration**.
4. In the Edit settings box, under **TS Session Broker**, double-click **Member of farm** in **TS Session Broker**.
5. On the **TS Session Broker** tab, click to select the **Join a farm in TS Session Broker** check box.
6. In the **TS Session Broker server name or IP address** box, type the name or the IP address of the TS Session Broker server.
7. In the **Farm name in TS Session Broker** box, type the DNS name or IP address of the domain server where the Terminal Services Session Directory service is running. Note that the server name must be a valid server name, and you cannot leave this box empty. (Also, the server name cannot be localhost or 127.0.0.1.)
8. Ensure that the **Participate in Session Broker Load-Balancing** check box is *not* enabled.
9. Ensure that the **Use IP address redirection** check box is enabled.
10. Click **OK** to save your settings.

For MSTS 2003 Session Directory configurations:

1. Click **Start > Administrative Tools > Terminal Services Configuration**.
2. In the console tree, click **Server Settings**.
3. In the **Details** pane, right-click **Session Directory**, and then click **Properties**.
4. Click to select the **Join session directory** check box.
5. In the **Cluster name** box, type the name of the terminal server cluster for this terminal server. This *must* be the same on all MSTS servers.

Deployment Guide

6. In the **Session directory server name** box, type the DNS name or IP address of the domain server where the Terminal Services Session Directory service is running. Note that the server name must be a valid server name, and you cannot leave this box empty. (Also, the server name cannot be localhost or 127.0.0.1.)
7. Verify that the **IP Address redirection** check box is enabled.
8. Click **OK** to save your settings.

Configuration Example 3: MSTS/Equalizer Deployment with Token Redirect Session Services

Overview

While IP redirection is the preferred setting for Session Broker, Equalizer can also work with a Session Broker that uses token redirection; the setup must be the same as **Configuration Example 2** with the following exceptions:

The sticky time on the cluster must be set to a non-zero value, and the session time on the MSTS servers must be set to the same value - as in Configuration Example 1. This means that the Token Redirection configuration suffers from the disadvantages outlined for Configuration Example 1 -- i.e., the client connection may end up getting load balanced only on initial connection, and will be 'stuck' to a server until the sticky time expires.

Equalizer makes a TCP connection to the server on the server port (3389) to determine if it is up, but by default does not validate that an RDP connection can be made.

This can lead to a situation where Equalizer thinks a server with a sticky record is up, but if the server is not able to make an RDP connection (e.g., the server is up but for some reason can't make an RDP connection), the client essentially gets 'stuck' on the server until the server can connect, until the sticky/session timers expire, or until the session is manually deleted from the Terminal Manager on the appropriate server.

To eliminate this possibility, the terminal servers should be set to End instead of Disconnect broken sessions, and an Equalizer server agent can be deployed to ensure that RDP is working in order for a server to be considered up.

Deployment Steps

The steps required to set up the example configuration involve defining the Equalizer cluster and the associated parameters, defining the servers and server settings, and deploying the server agent on each MSTS server:

Deployment Guide

Equalizer Cluster

1. Define the Terminal Services virtual cluster
2. Disable the **spoof** option on the cluster and set a **sticky time**.
3. Setting the LB algorithm for the MSTs Cluster

Cluster Servers

4. Add real Terminal Servers to virtual cluster
5. Set max connection value for each Terminal Server

MS Terminal Servers

6. Set the **session time** on each Terminal Server to any value desired, or leave the default (Never).
7. Set the **maximum connections** value on each Terminal Server to the same value used for **max connections** on the corresponding server definition on Equalizer.
8. Configure each Terminal Server to:
 - join a Session Directory/Session Broker server farm
 - *Disable IP redirection*. This enables **token redirection**.
9. Set each Terminal Server to **End** broken sessions instead of **Disconnect**.
10. Deploy a server agent on each Terminal Server.

Equalizer Cluster

11. Configure the Equalizer cluster for the server agent.

Step 1: Defining the Terminal Services virtual cluster

Follow the instructions in Step 1 of **Configuration Example 1**.

Step 2a: Set 'sticky time' for the Terminal Services Cluster

Follow the instructions in Step 2 of **Configuration Example 1**.

Step 2b: Disable 'spoof' for the Terminal Services Cluster

Follow the instructions in Step 2 of **Configuration Example 2**.

Step 3: Set LB algorithm for the Terminal Services Cluster

Follow the instructions in Step 3 of **Configuration Example 1**.

Step 4: Add real servers to Terminal Services Cluster

Follow the instructions in Step 4 of **Configuration Example 1**.

Step 5: Set Max Connections

Follow the instructions in Step 5 of **Configuration Example 1**.

Step 6: Set MS Terminal Server Session Time

Deployment Guide

Do one of the following:

- If you want to leave the default value of 'Never' for the session time, skip this step.
- If you want to set the session time to a lower value, follow the instructions in Step 6 of [Configuration Example 1](#).

Step 7: Set MS Terminal Server Max Connections

Follow the instructions in Step 7 of [Configuration Example 1](#).

Step 8: Configure MS Terminal Server for Token-based Session Reconnection

Follow the instructions in Step 8 of [Configuration Example 2](#), with this exception:

- *disable* the IP address redirection check box (this enables token-based redirection).

Step 9: Set MS Terminal Servers to End Broken Connections

1. On each MSTS server, open the Server Manager by clicking Start > Programs > Administrative Tools > Server Manager.
2. In the left panel of the Server Manager, click the plus sign next to Roles to expand the Roles list, then click the plus sign next to Terminal Services to expand that list.
3. Click on Terminal Services Configuration.
4. In the right panel, under Connections, right-click on the Connection Name (default is RDP-Tcp) and select Properties.
5. Click on the Sessions tab (see Figure 1.7)
6. Enable the Override user settings check box and then click the End session radio button.
7. Click OK to save your changes.

Step 10: Deploy a Server Agent on each Terminal Server

Server Agents can be written in any programming or scripting language supported on the server. The program must listen for connections on the port to which Equalizer will connect for the server agent probe (the default is 1510), and return a number between -2 and 100 when a connection is made.

For this example, we use a simple server agent written in perl that uses Windows command line utilities to determine if the system is listening for Terminal Server connections. On our system, we installed ActivePerl 5.10, but you can use any perl distribution that you choose. Be sure to install perl on every server on which you want to run the server agent, and also be sure that

Deployment Guide

the “.pl” file extension is associated with the perl executable that you install (this is done automatically by the ActivePerl install scripts).

The text of the server agent perl script is given in *Appendix A: Terminal Services Server Agent*. Lines 33 through 46 of the script determine the number to return to Equalizer using this process:

- 1) The output of the `qwinst` command is parsed (using `find`) for the string “Listen” - this indicates that there is at least one process on the system listening for terminal services connections.
- 2) If “Listen” is not found, then no terminal services process is listening for connections and we set the agent response to “-1”, indicating that the service is down. The server will be marked down on Equalizer.
- 3) If “Listen” is found, the agent does a rough load calculation to return to the server. In the example script, we assume that the `max connections` setting of the server on Equalizer and in MSTs is 50 sessions, so there may be from 0 to 50 sessions active at any time. The script counts the number of Active sessions, multiplies the result by two and adds 1. This results in a number between 1 and 101, so there is a check to make sure the response sent is between 1 and 100 (the supported server agent return values on Equalizer). This approximation of the server load is returned to Equalizer and used by Equalizer’s load balancing algorithms.

To install and run the server agent on your MSTs servers, do the following on each server:

1. Copy and paste the server agent perl code from Appendix A into a file named `svrgnt-msts.pl` on the MSTs server.
2. Open the Server Manager utility, and navigate to **Configuration > Task Scheduler**. Use the **Task Scheduler** to schedule the script to run as follows:
 - run the script every time the system starts
 - if the task fails to run, restart it every five minutes
 - allow the task to be started manuallyThe Task Scheduler is a bit different depending on which Windows Server version you are using, so check the documentation on your system for complete instructions.
3. In the **Task Scheduler** window, right click on the new task and click **Run** to start the task manually. A window will open showing the startup output of the command “**Server agent started on port 1510**”. Do *not* close this window or the agent will stop running. The script output will also show when Equalizer connects to the agent, after the next step is performed.

Step 11: Configure the Equalizer cluster for the server agent

1. In the left pane of the Equalizer Administrative Interface, click on the Equalizer’s system name (default: Equalizer).

Deployment Guide

2. In the right pane, open the **Probes** tab.
3. Enable the **require agent response** check box.
4. Click **commit** to save the change.
5. In the left pane of the Equalizer Administrative Interface, click on the name of the MSTS Cluster.
6. In the right pane, open the **Configuration > Probes** tab.
7. In the agent type box, enable the **server agent** radio button.
8. Click **commit** to save the change.
9. To avoid consuming a Terminal Services license for TCP probes, click on the name of each server in the cluster, enable **don't probe**, and click **commit**.

To confirm that Equalizer is communicating with the server agent, click on the name of a server in the left frame, open the **Reporting > Plots** tab for the server, and enable the **Server Agent** check box. The chart should show the same **response** values reported by the server agent in its output window on the MSTS server (see the previous Step 10, above).

For Additional Help

If you have additional support questions, Customer Support contact information is available from the **Support** link on our main web page at <http://www.coyotepoint.com>.

If you haven't already, be sure to register for access to the Coyote Point Support Portal at: <http://support.coyotepoint.com> for access to the Coyote Den Forum, FAQs, Product Documentation, Release Notes and other valuable support resources.

Appendix A: Terminal Services Server Agent

The sample Server Agent included in this appendix is intended to be installed and run on a Windows Terminal Services enabled server to provide accurate probe feedback to Equalizer. It can be used in any Equalizer / Terminal Services configuration to avoid probing the Terminal Services port on the server and, by doing so, to avoid consuming a Terminal Services license. The general procedure is to:

- Turn TCP and ACV probing off for the server (enable "don't probe" on the server's configuration tab).
- Run a program on the terminal server (a "server agent") to return server status information. The server agent can be configured to return simple up/down status, or an indication of the load on the server. A sample server agent is below. By default, server agents use port 1510, but this can be changed.
- Enable the cluster "require agent response" flag to require a server agent response in order to mark the server up.

The complete procedure is provided in Step 10 and 11 on pages 16 and 17.

Sample Server Agent

```

1.  #-----
2.  #  srvrgnt-msts.pl
3.  #-----
4.  #(c) Copyright 2009 Coyote Point Systems Inc.
5.
6.  use strict;
7.  use Socket;
8.
9.  # YOU MUST SET maxconn to the same value used for maximum connections on
10. # the MS Terminal Server on which this agent will be installed.
11. # As delivered, this script uses an arbitrary default of 75.
12. my $maxconn=75;
13.
14. # YOU MUST SET port to the same value used for server agent port on
15. # Equalizer. The Equalizer default is 1510
16. my $port = 1510;
17.
18. # create a socket and set the options, set up listen port
19. my $proto = getprotobyname('tcp');
20. socket(SERVER, PF_INET, SOCK_STREAM, $proto) or die "socket: $!";
21. setsockopt(SERVER, SOL_SOCKET, SO_REUSEADDR, 1) or die "setsock: $!";
22. my $paddr = sockaddr_in($port, INADDR_ANY);
23.
24. # bind to the port, then listen on it; print diagnostic
25. bind(SERVER, $paddr) or die "bind: $!";
26. listen(SERVER, SOMAXCONN) or die "listen: $!";

```

Deployment Guide

```
27. print "Server agent started on port $port\n";
28.
29. # start a loop to accept Equalizer connections
30. my $client_addr;
31. while ( $client_addr = accept(CLIENT, SERVER)) {
32.
33. # find out who connected
34. my ( $client_port, $client_ip) = sockaddr_in($client_addr);
35. my $client_ipnum = inet_ntoa($client_ip);
36. # print who has connected as diagnostic
37. print "Connection from: [$client_ipnum]\n" ;
38.
39. # get the server agent response value; default -2
40. # Equalizer interprets the response value as below:
41. # -2 = no status
42. # -1 = service unavailable
43. # 0 = heavy load
44. # 100 = light load
45. my $response = -2;
46. # need to count the number of active connections in the qwinsta
    output
47. my $currconn = 0;
48. # need to determine the multiplier or divisor to use when
    calculating
49. # the response
50. my $mp = 1;
51. my $dv = 1;
52.
53. # see if there is an MSTTS Listen process running
54. my $lstatus = system('qwinsta | find "Listen" > NUL');
55. # if $lstatus = 0 then a Listen process was found
56. # if a Listen process was found, count the Active entries and
57. # calculate an agent response value between -1 and 100;
58. # if a Listen process is not found, return -1 (service down)
59. if ( $lstatus == 0 ) {
60.     $currconn = `qwinsta | find /C "Active"`;
61.     if ( $maxconn == 100 ) { $response = $currconn;
62.         } elseif ( $maxconn < 100 ) {
63.         $mp = ( 100 / $maxconn );
64.         $response = ( $currconn * $mp );
65.         } elseif ( $maxconn > 100 ) {
66.         $dv = ( $maxconn / 100 );
67.         $response = ( $currconn / $dv );
68.         }
69.     if ( $response < 0 ) { $response = 0; }
70.     else { if ( $response > 100 ) { $response = 100; }
71.     }
72.     $response = ( 100 - $response );
73. } else { $response = -1; }
74. # print lstatus and response as diagnostic
75. print "lstatus = ", $lstatus/256, "\n", "currconn =
    ", $currconn, "response = ", $response, "\n" ;
76.
77. # send the response value to Equalizer
```

Deployment Guide

```
78. # ***do not comment out the print statement below***
79. print CLIENT $response ;
80.
81. # close the connection; return to listening
82. close CLIENT;
83. }
```

About Coyote Point

Coyote Point Systems Inc., the original load balancing technology pioneer, is a recognized leader in delivering affordable solutions for high application availability and accelerated application performance. With over a decade of experience, Coyote Point has delivered the eighth generation of its award-winning Equalizer Series platform. Equalizer Series load balancing and acceleration systems provide the industry's foremost combination of performance, affordability and ease of use. Over 8,500 customers worldwide have deployed Equalizer Series systems to ensure non-stop operations, accelerated delivery and on-demand scalability of business-critical applications and Websites. Coyote Point is headquartered in San Jose, CA and works with leading channel partners in the U.S., Canada, Europe, Asia, Australia and Africa.

Visit our website at <http://www.coyotepoint.com>.