

Siebel CRM 8.0 Deployment Guide

Utilizing the Acceleration and Optimization Features of Citrix® NetScaler®

A TECHNICAL GUIDE





INTRODUCTION

Enterprise operations must manage customer interactions across a variety of communications channels often on a global basis. Customer Relationship Management (CRM) applications form the backbone of such exchanges. These solutions enable customer-facing business processes such as lead generation and cross-sell, opportunity management, forecasting and quoting, sales support and service, collaborative channel management, eCommerce, customer data analytics, and customer data management. Various estimates show worldwide revenue from CRM approaching \$11 Billion in a few years. While the clear majority of this total is aimed at large enterprise deployments, SMB opportunities are growing at a faster pace.

These applications may be focused on CRM but they are highly integrated software products with a host of interrelated functions. One of the leading enterprise class solutions is Oracle's Siebel CRM. This suite is composed of several elements including Business Analytics, Partner Relationship Management, Call Center and Service, Customer Order Management, and Enterprise Marketing. Tight integration is demanded to provide a simplified customer interaction with the organization.

This document is intended to be a guideline for deploying Citrix NetScaler Application Deliver solutions with Siebel CRM 8.0. This guide will provide steps for improving the Siebel end user experience by utilizing Citrix NetScaler's Load Balancing, acceleration, and optimization features. The configuration examples are extractions from a Citrix test lab validated by Oracle. This guide is not designed to replace existing Citrix NetScaler Implementation and Configuration Guides (ICG) or Oracle's Siebel planning and deployment documentation.

TABLE OF CONTENTS

Prerequisites	5
Citrix Application Delivery and Optimization Features	6
Application-Layer Switching	6
TCP/IP Multiplexing and Connection Management	6
Web Compression	6
Application Data Caching	6
Server Load Balancing	6
Siebel 8 Load Balancing Application Notes	7
NetScaler Configuration Notes	10
NetScaler Configuration	11
Deployment Model: NetScaler One-Arm Mode	11
Connecting to Citrix NetScaler	12
Configuring Citrix NetScaler Global Features	12
Global Policy Expressions	14
Create Policy Expressions	15
Citrix NetScaler Compression	26
Configuring Compression for Siebel	26
Citrix NetScaler Static Caching	29
Citrix NetScaler Load Balancing	35
Server Load Balancing	38
Configuring Siebel 8 Web Virtual Server	41
Configuring Siebel 8 Application Virtual Server	42
Session Persistence for Siebel	43
Conclusion	44

PREREQUISITES

- Proficiency with deployment of Oracle's Siebel 8.0 and its components.
- Knowledgeable of Citrix NetScaler Installation and Configuration Guide (ICG) Volume 1 & 2. There are several sections in this document that refer to the ICG for further discussion and configuration considerations.
- Intermediate or Advanced knowledge of Networking, and Web technologies.
- NetScaler running version 7.0 or higher used in this deployment example.

CITRIX APPLICATION DELIVERY AND OPTIMIZATION FEATURES

Application-Layer Switching

Application-layer switching capabilities provides application content distribution among multiple application servers, ensuring increased application performance with fail-over support for business continuity in an Microsoft SharePoint Services environment. Citrix Request Switching® ensures even traffic distribution irrespective of individual user demands.

TCP/IP Multiplexing and Connection Management

TCP/IP multiplexing and connection management dramatically reduces the number of TCP connections each SharePoint server is required to manage, allowing organizations to reduce their server infrastructure or serve a significantly larger number of clients, depending on need. NetScaler optimizes the use of standard Internet protocols by multiplexing requests from a very large number of users to a much smaller number of servers via persistent connections between clients and servers.

Web Compression

AppCompress™ improves performance by reducing the amount of data sent from Web servers to browsers. Redundant data is removed from messages sent to clients, and then compression software that is built into virtually all Web browsers recreates the data exactly as it was created by the server. This makes Web compression transparent to all Web-facing applications.

Application Data Caching

AppCache™ improves performance by retaining frequently accessed transaction data and serving it in response to repeated requests from the client rather than application servers. This accelerates response times and also reduces the load on Web, middleware and database servers.

Server Load Balancing

Suites with numerous applications require multiple servers to fulfill requests for data, applications and web content. Citrix NetScaler provides unparalleled capabilities to ensure application accessibility. Layer 4 based server load balancing automatically directs client requests to proper back-end application, database or web servers. Such servers and applications can be grouped by server, services or service groups. The latter allow Siebel software elements to be mapped to one or more specific servers and treated as a logical entity by Citrix NetScaler. This enables each of the applications to be distinguished such that if one is undergoing an upgrade or other maintenance the other applications are not affected.

SIEBEL 8 LOAD BALANCING APPLICATION NOTES

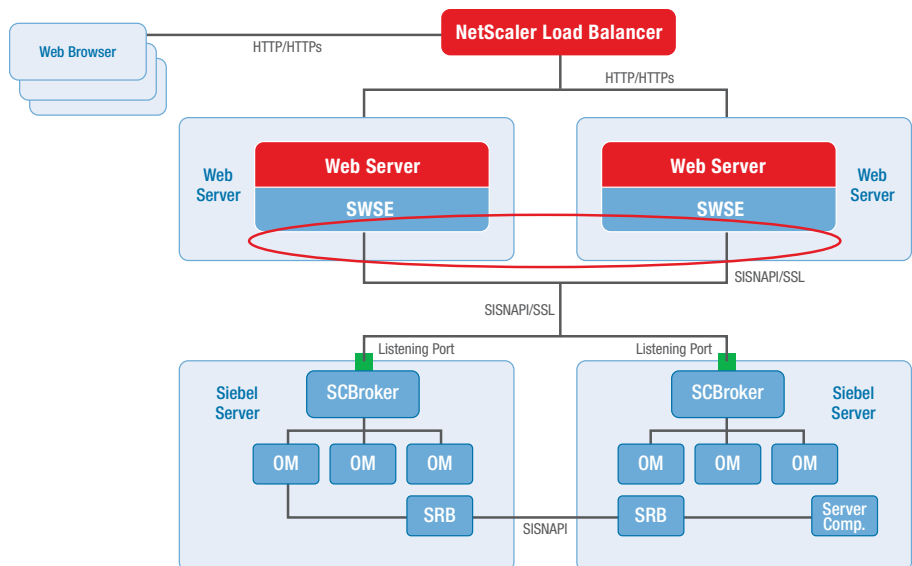
With Siebel 8.0, you can choose between 2 deployment options for load balancing with NetScaler:

Option 1

- NetScaler receives requests from client browsers and load balances them across Siebel Web servers. Load balancing Siebel web servers do not require any specific configuration changes on the web server. The NetScaler appliance can be placed in front of the Web servers and can load balance requests.
- Siebel Load balancing for Application servers. In this case, each Web server is directly tied to an application server.

NOTE: Siebel Server Load Balancing for Application servers is typically done after the Siebel Servers and Database Server have been installed, but can be done before the Web servers are installed. The Siebel Load Balancing instance resides in the Siebel Web Server Extension (SWSE) on the Web Servers. It allows each instance of SWSE to distribute connection requests to multiple application servers in a round-robin fashion.

Typically, the Siebel administrator will generate the load balancing configuration file by logging into the Server Manager and type “generate lbconfig”. This generates a load balancing configuration file in the SIEBEL_INSTALLATION_ROOT/Admin directory.

Logical Architecture for Option 1

Option 2

- NetScaler receives requests from client browsers and load balances across Siebel Web servers.
- NetScaler receives application server requests from all Siebel Web servers and load balances to multiple Siebel Application servers.

NOTE: To implement a hardware based load balancer for Siebel Application servers, a couple of configuration changes on the Siebel Web servers are required.

1. First, update the eapps.cfg file to disable Siebel Load Balancing by changing the setting below as follows:

EnableVirtualHosts = False

2. Next, modify the Object Manager connect string so it points to the Virtual IP and Port. For the load balanced Object Manager, the connect string must have the format:

**ConnectString =
siebel.TCPIP.None.None://<VirtualIP>:<VirtualPort>/<Siebel
Enterprise Name>/<Alias of the Object Manager>**

Where:

<VirtualIP> is the IP address of the Virtual Server specified in NetScaler. This VIP will be designated as the Application Server.

<VirtualPort> is the Port Number, or Service defined in the Virtual Server definition. The default port is 2321.

<Siebel Enterprise Name> is the name of the Siebel Enterprise in which the load balanced Siebel Servers reside.

<Alias of the Object Manager> is the alias of the Load Balanced Object Manager.

Example:

Locate the file in:

D:\SBA80\SWESpp\BIN\eapps.cfg

Original eapps.cfg file

EnableVirtualHosts = **true**

[/callcenter_enu]

ConnectString =

siebel.TCPIP.None.None://VirtualServer/SBA_80/SCCObjMgr_enu

Updated changes to the eapps.cfg

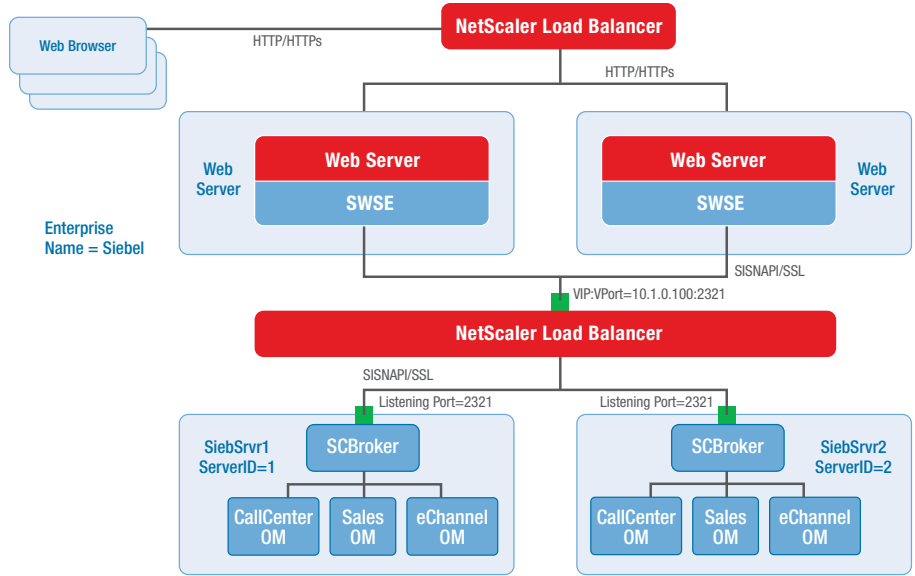
EnableVirtualHosts = **false**

[/callcenter_enu]

ConnectString =

siebel.TCPIP.None.None://172.16.10.243:2321/SBA_80/SCCObjMgr_enu

Logical Architecture for Option 2



NETSCALER CONFIGURATION NOTES

Pre-Configuration Checks

Before configuring NetScaler to load balance Siebel Servers, check to ensure NetScaler is set up properly to route TCP/IP and HTTP traffic. The following steps are generally required:

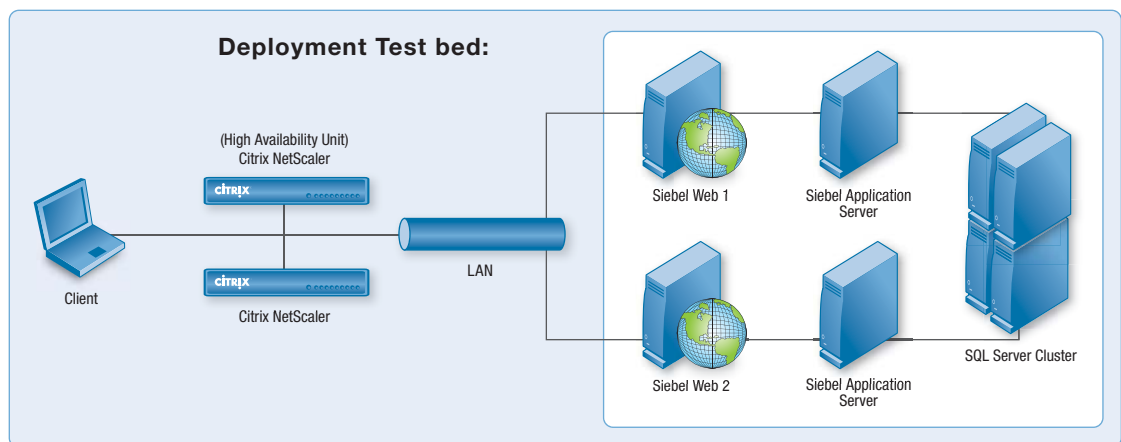
1. Plan out the network topology with NetScaler. NetScaler itself is a network entity, and may require load balanced servers to reside in the same subnet. Therefore, NetScaler affects how the network is laid out. You also need to plan out what Virtual IP address to use, which Port to configure, etc.
2. Install NetScaler in the Data Center and hook up the network cables to the switches and hubs.
3. Initialize NetScaler with vendor-provided license keys, and assign static IP addresses.
4. Setup failover NetScaler boxes to ensure high availability.
5. Configure the Network Gateway, Subnet, and all other networking parameters for the Virtual LANs (VLANs) supported by NetScaler. This is typically a critical step, as it determines what IP addresses Siebel Servers can use while being load balanced.
6. Set up machines that will host the Siebel application, and configure the TCP/IP properties for these machines. Test out basic TCP/IP connectivity between NetScaler and Siebel servers.
7. There are several Siebel-specific restrictions when planning out the networking topology:
 - a) Each URL on each instance of the SWSE can only point to one Object Manager (OM) connect string. This means one virtual IP (VIP) and virtual port combination per URL for a particular SWSE.
 - b) Each URL and OM connect string typically corresponds to one Siebel Application.
 - c) Typically, one application has one VIP, although a VIP can be shared across multiple applications. This allows all servers hosting an application to be load balanced together. For example, for a customer running Call Center, Sales, and the eChannel applications, the following partitioning schemes follows the above rule:
 - VIP1: Call Center, Sales, and eChannel
 - Call Center and Sales. VIP2: eChannel
 - Call Center. VIP2: Sales. VIP3: eChannel
 - d) The following partitioning scheme does not:
 - VIP1: Call Center, Sales. VIP2: Sales, eChannel
 - Sales OM connect string can only point to one VIP
 - e) It is possible to configure multiple VIPs for one Siebel Application, but the Web servers must be partitioned accordingly. Typically, multiple Virtual IP addresses are unnecessary.

NETSCALER CONFIGURATION

Deployment Model: NetScaler One-Arm Mode

NetScalers can be deployed as a pair to provide high availability for the Siebel application. NetScalers in One-Arm mode can be transparently integrated into the Siebel environment without any physical changes to the existing network infrastructure. Siebel Applications themselves do not have any specific requirements around the implementation topology of the Load Balancers. Overall network topology and layout, on the other hand, dictates where and how Load Balancers can be deployed. The Siebel administrator and Network administrator will require collaborative efforts in ensuring successful TCP/IP connectivity and DNS name resolution between the Siebel servers and Virtual IPs (VIPs) configured on NetScaler.

For further discussion on the various NetScaler deployment models, refer to section 2.2.1 (Planning the Deployment) of the NetScaler ICG guide Volume 1.



NetScaler Platform and Version

Hardware: NetScaler 7000

Version: 7.0

Siebel Environment

Server Hardware: Dual Xeon processors, 4 GB RAM

Software:

> Siebel 8.0 for Windows

NOTE: It is recommended that the existing configuration be saved before beginning the following procedures in this deployment guide. See section 3.0 of the NetScaler ICG guide Volume 1 for instructions on how to save settings.

Connecting to Citrix NetScaler

1. To access the Configuration Utility from the browser, type the system's default IP address in the address bar of the Web browser:

http:// <NetScaler Management IP address>

The system homepage is displayed.

2. To launch the Configuration Utility, click the Applet Client or Web Start Client hyperlink on the right-hand side of the Configuration Utility label. The login page is displayed.
3. At the login prompt, type the user name nsroot and the password nsroot and click Login. The Setup Wizard is displayed with the Configuration Utility in the background.

Configuring Citrix NetScaler Global Features

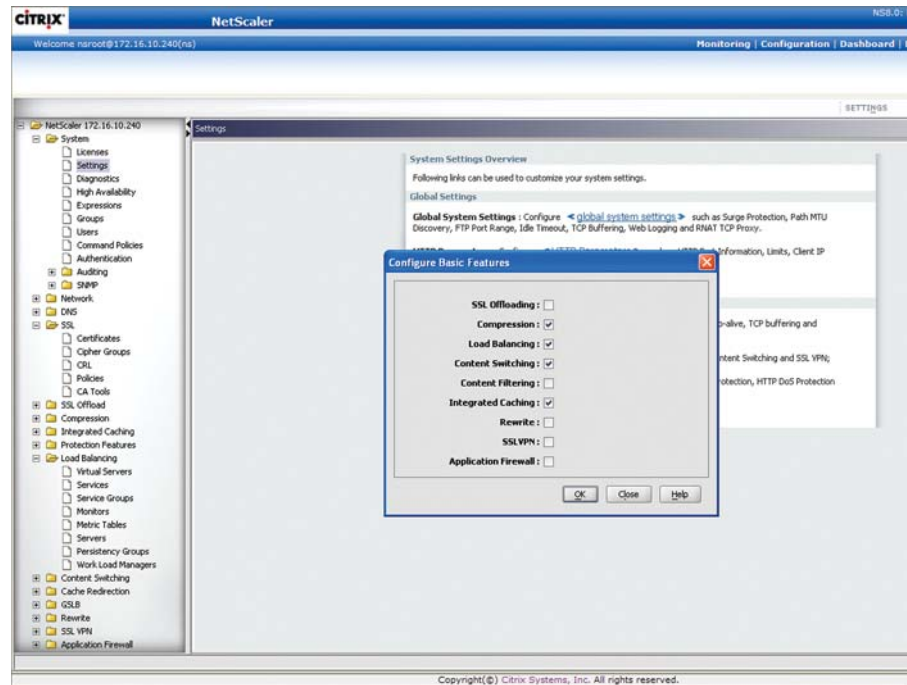
The following NetScaler features are required for the Siebel deployment:

- Compression
- Load Balancing
- Integrated Caching

Go to the Navigation Panel (left side of the main NetScaler Configuration Utility) and expand the **System** Node.

Click **Basic Features**.

Select the following feature boxes and click **OK**: Compression, Load Balancing, Content Switching and Integrated Caching.



NOTE: While NetScaler offers both GUI-based and command line interface configuration tools, this guide will solely focus on the GUI-based method.

Global Policy Expressions

A Policy Expression is a set of conditions that can be applied on content entering the NetScaler system. Expressions represent one or more of these conditions and make up a Policy Expression. The Policy Expressions are shared among the NetScaler features. The NetScaler Compression, Integrated Caching, and Content Switching features enabled for Siebel are controlled by Policy Expressions. These Policy Expressions can be created through various windows within the NetScaler Configuration Utility. They can be created within the **Feature** node or at the **System** (Global) node of the NetScaler Configuration Utility. The System node represents a global repository for Policy Expressions and can provide a benefit to the system administrator's management responsibilities for all the expressions.

For further discussion on Policy Expressions, refer to Chapter 15 of the NetScaler Installation and Configuration Guide Volume 1.

The following steps will illustrate the creation of the Policy Expressions within the **System** node.

For Static Caching, the names of the following Policy Expressions will be created for the following HTTP objects:

- gifs (images)**
- jpeg / jpg (images)**
- js (javascript)**
- css (Content Style Sheets)**

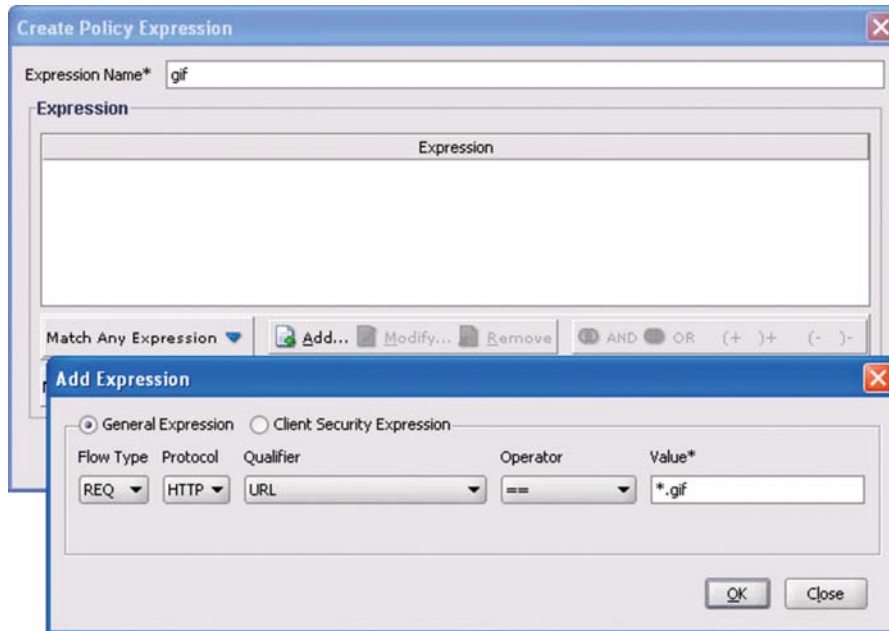
For increased compression, the names of the following Policy Expressions will be added:

- js (javascript)**
- js_content_type (javascript)**

Policy expression for gifs

In the Expression **Name*** field, enter the name for this Expression.

In the example, the Expression name is **gif**.

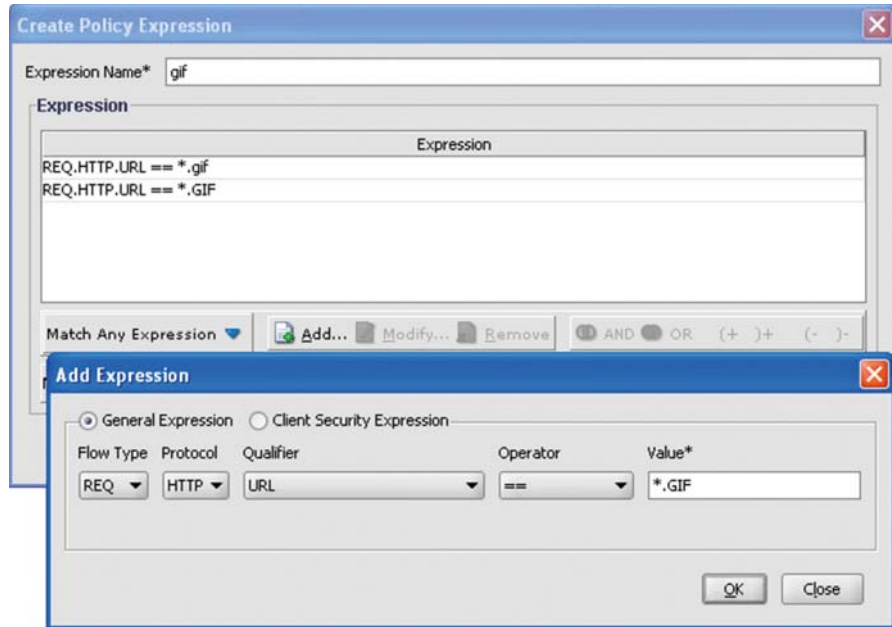


Select **URL** for the Qualifier.

Type ***.gif** in the **Value*** field.

Click **OK** to create the Expression in the box. This is shown in the Expression box and shown as **REQ.HTTP.URL== *.gif**.

Repeat the above step for a ***.GIF** expression.



Once the ***.GIF** expression has been created, Close out of the Add Expression sub-window and Click **Create**.

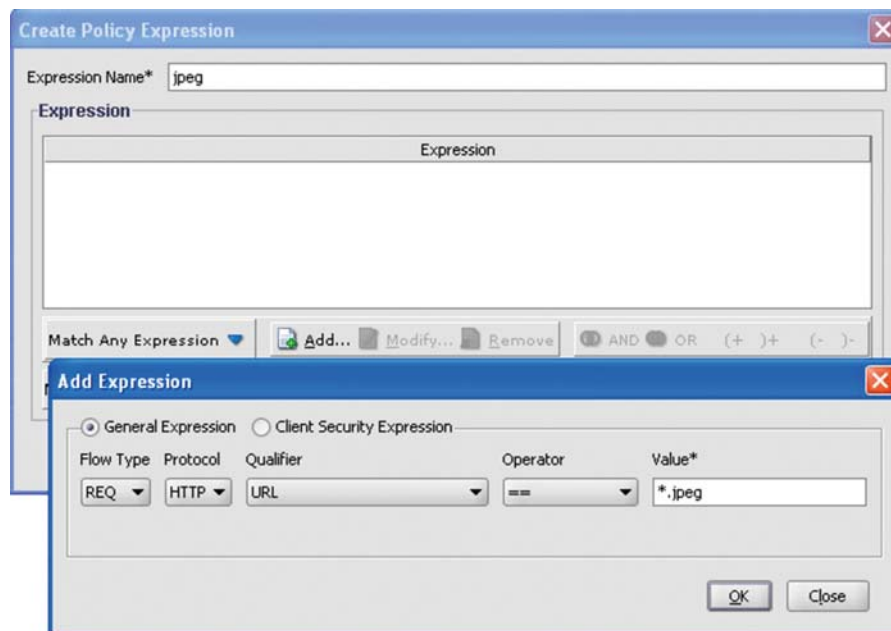
The expressions **REQ.HTTP.URL==*.gif** and **REQ.HTTP.URL==*.GIF** will now have formed the Policy Expression **gif**.

These expressions will identify the gif file extension within the URL.

Policy Expression for jpeg

In the Expression **Name*** field, enter the name for this Expression.

In the example, the Expression name is **jpeg**.

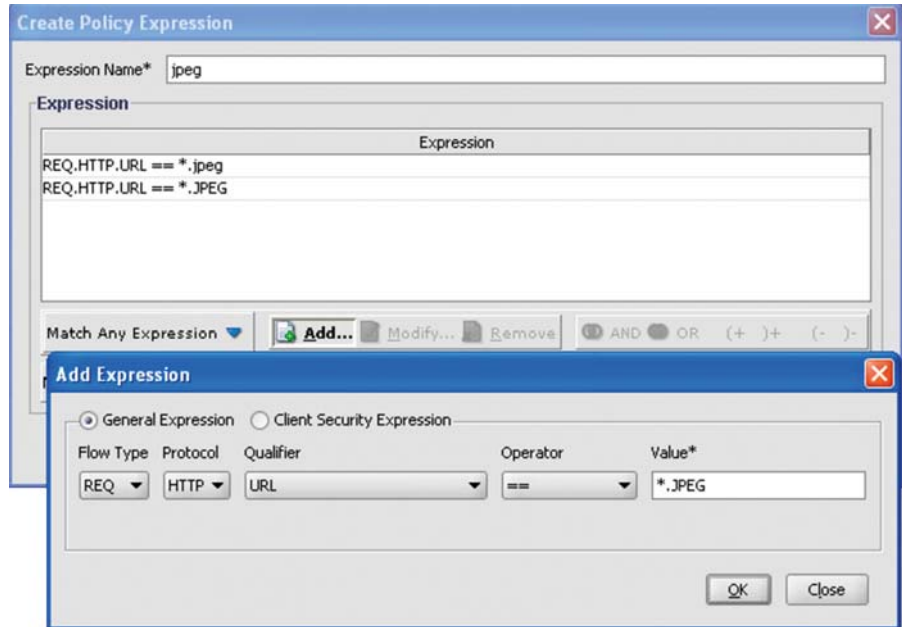


Select **URL** for the Qualifier.

Type ***.jpeg** in the **Value*** field.

Click **OK** to create the Expression in the box. This is shown in the Expression box and shown as **REQ.HTTP.URL== *.jpeg**.

Repeat the above step for a ***.JPEG** expression.



Once the ***.JPEG** expression has been created, Close out of the Add Expression sub-window and Click **Create**.

The expressions **REQ.HTTP.URL==*.jpeg** and **REQ.HTTP.URL==*.JPEG** will now have formed the Policy Expression **gif**.

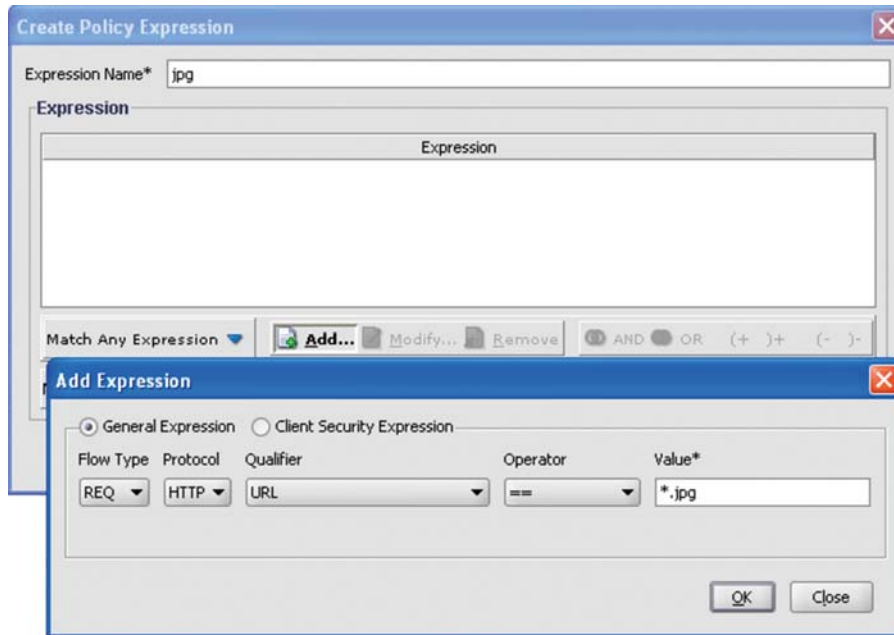
The expressions will identify the jpeg file extension within the URL.

The creation of these expressions will identify an http/https object with the jpeg and JPEG file extension within the URL and will later be used for Static Caching policies.

Policy Expression for jpg

In the Expression **Name*** field, enter the name for this Expression.

In the example, the Expression name is **jpg**.

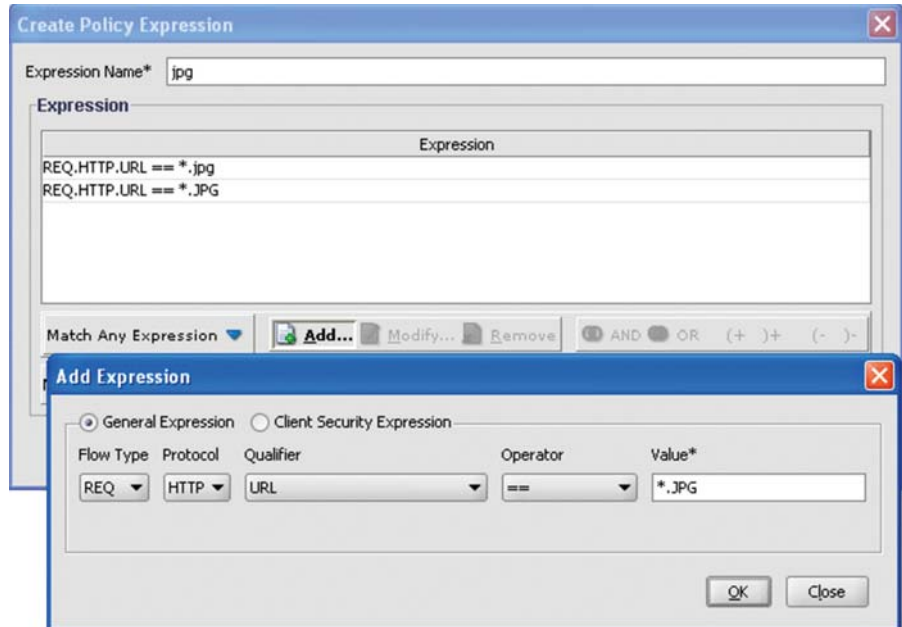


Select **URL** for the Qualifier.

Type ***.jpg** in the **Value*** field.

Click **OK** to create the Expression in the box. This is shown in the Expression box and shown as **REQ.HTTP.URL==*.jpg**.

Repeat the above step for a **.JPG** expression.



Once the ***.JPG** expression has been created, Close out of the Add Expression sub-window and Click **Create**.

The expressions **REQ.HTTP.URL==*.jpg** and **REQ.HTTP.URL==*.JPG** will now have formed the Policy Expression **jpg**.

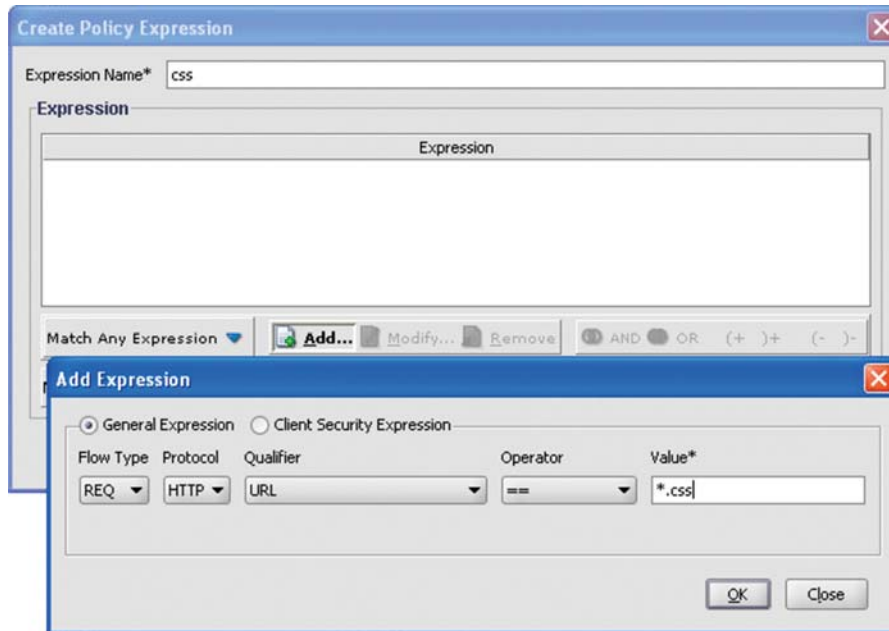
These expressions will identify the jpg file extension within the URL.

The creation of these expressions will identify an http/https object with the jpg and JPG file extension within the URL and will later be used for Static Caching policies.

Policy Expression for css

In the Expression **Name*** field, enter the name for this Expression.

In the example, the Expression name is **css**.

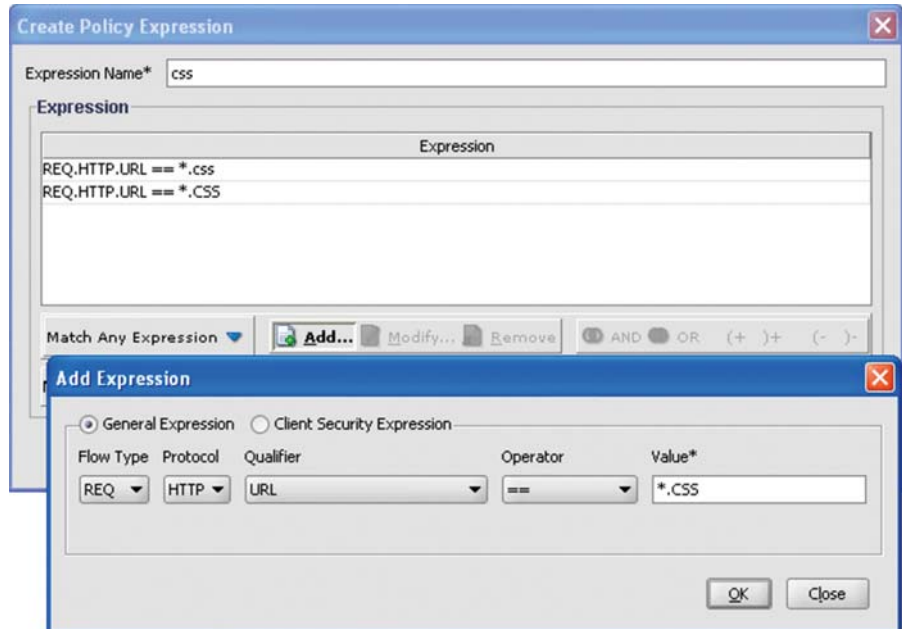


Select **URL** for the Qualifier.

Type ***.css** in the **Value*** field.

Click **OK** to create the Expression in the box. This is shown in the Expression box and shown as **REQ.HTTP.URL== *.css**.

Repeat the above step for a ***.CSS** expression.



Once the ***.JPG** expression has been created, Close out of the Add Expression sub-window and Click **Create**.

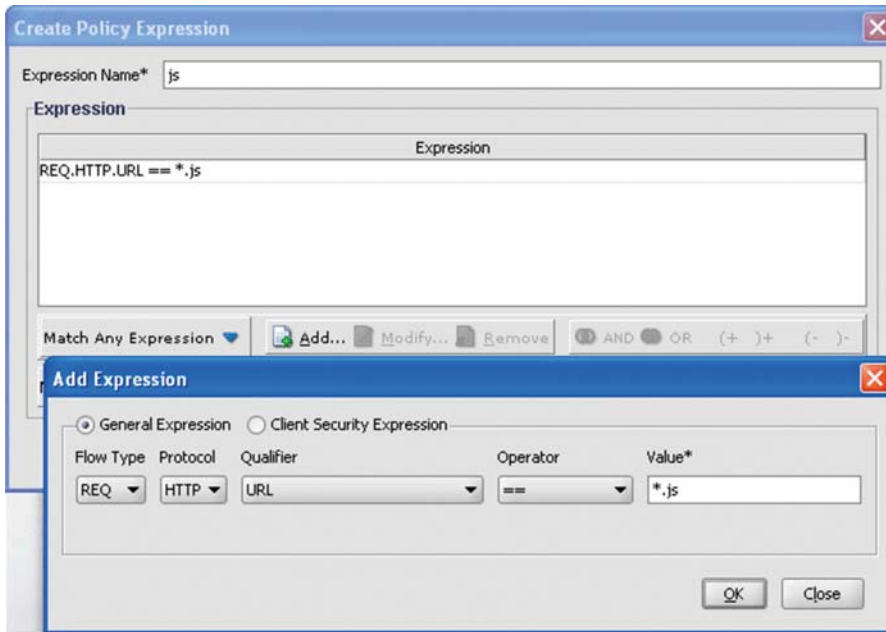
The expressions **REQ.HTTP.URL==*.css** and **REQ.HTTP.URL==*.CSS** will now have formed the Policy Expression **css**.

The creation of these expressions will identify an http/https object with the css and CSS file extension within the URL and will later be used for Static Caching policies.

Policy Expression for js

In the Expression **Name*** field, enter the name for this Expression.
In the example, the Expression name is **js**.

Click **Add**.



Once the **.js** expression has been created, close out the Add Expression sub-window and Click **Create**.

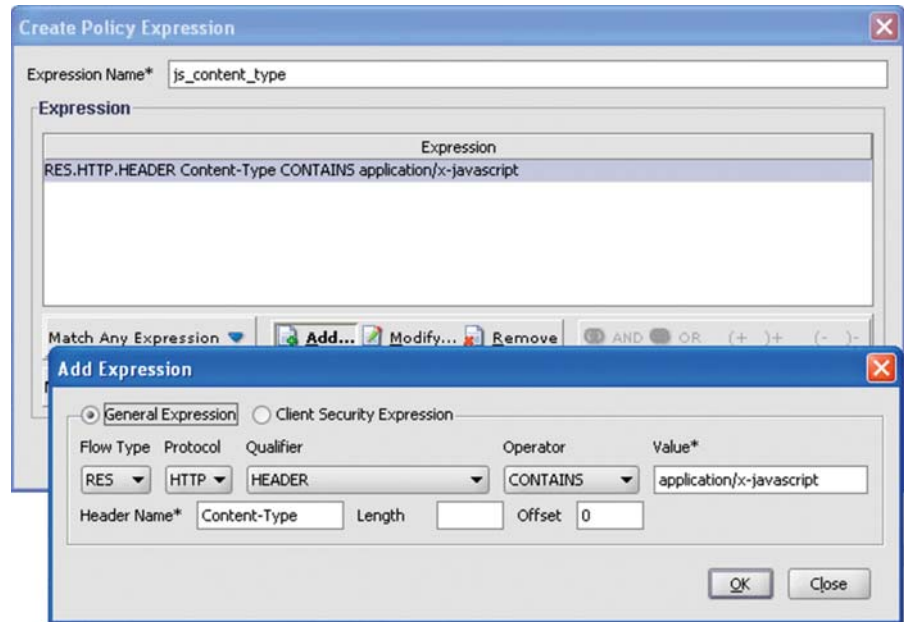
This expression will identify the javascript file extension within the URL.

The creation of this expression will identify an http/https object with the js file extension within the URL and will later be used for Static Caching and Compression policies.

Policy Expression for js_content_type

In the Expression **Name*** field, enter the name for this Expression. In the example, the Expression name is **js_content_type**.

Click **Add**.



Select **RES** for the Flow Type.

Select **Header** for the Qualifier.

Select **Contains** for the Operator.

Type **application/x-javascript** in the **Value*** field.

Click **OK** to create the Expression in the box. This is shown in the Expression box and shown as **RES.HTTP.HEADER Content-Type CONTAINS application/x-javascript**.

This expression will identify the javascript MIME type within the HTTP header and will later be used for Static Caching and Compression policies.

CITRIX NETSCALER COMPRESSION

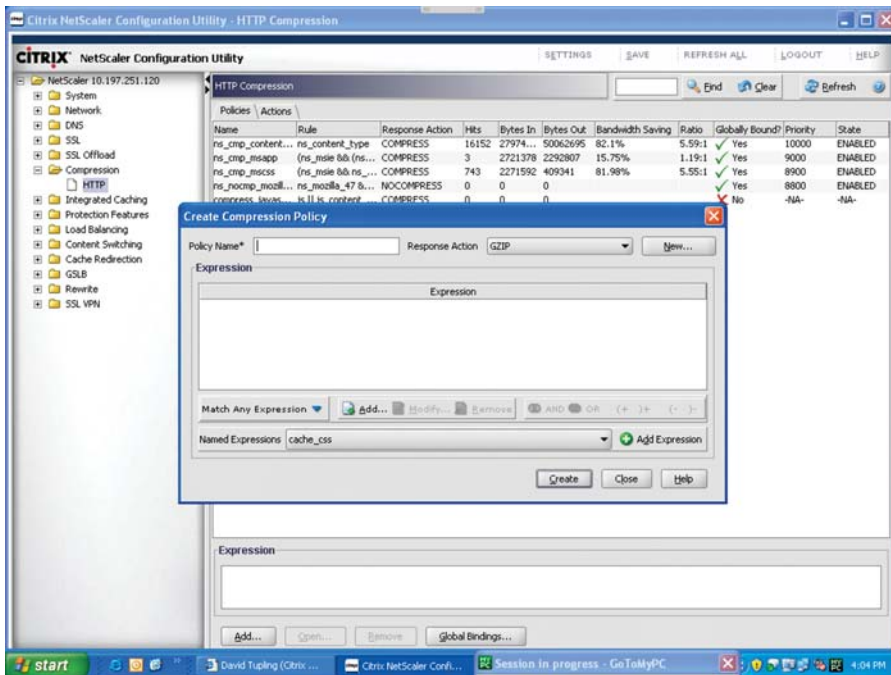
The NetScaler system implements lossless compression that can be interpreted by popular browsers like Internet Explorer, Netscape, and AOL. It can compress payloads up to a ratio of 4:1. By default, the system compresses text/HTML and text/* MIME formats for all browsers. The NetScaler system compresses traffic based on the format supported by the browser. While the NetScaler system can compress content generated by most CGI applications, by default it does not compress client side javascript traffic.

Like many other web applications, Siebel incorporates the use of javascript within the various Siebel tasks.

Configuring Compression for Siebel

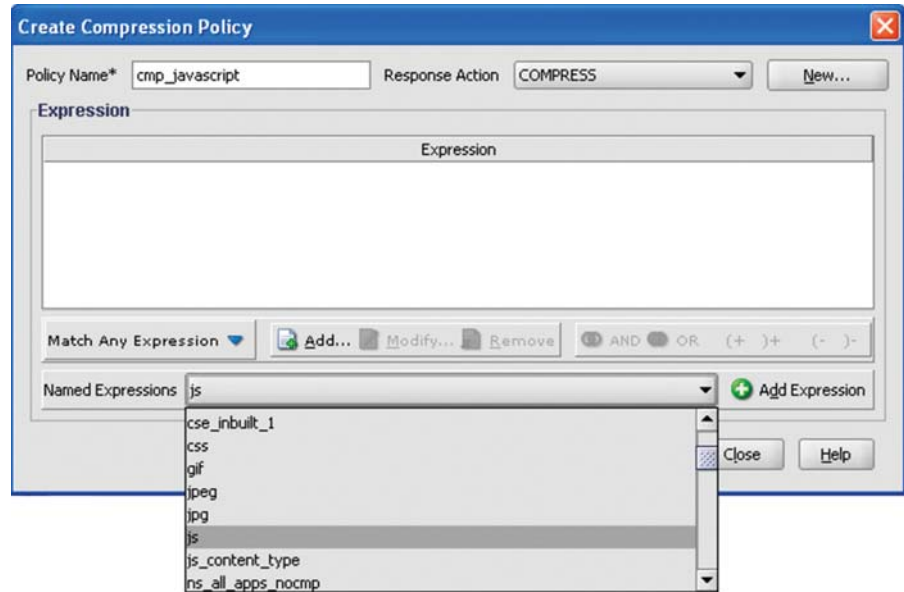
Go to the **Navigation Panel** (left side of the main NetScaler Configuration Utility) and expand the **Compression** Node.

Select the **HTTP** sub-node and click **Add**.



Type **cmp_javascript** in the **Policy Name*** field and select the **Response Action** pull down box to **COMPRESS**.

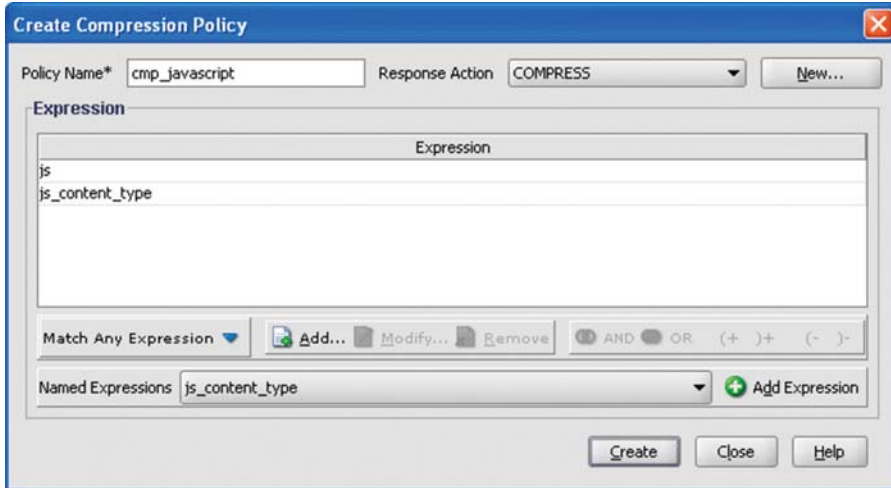
In the **Named Expressions** pull down box, select **js** (Policy Expression created in the **System** Node).



Add another Named Expression to the compression policy.

In the **Named Expressions** pull down box, select **js_content_type** (Policy Expression created in the **System** Node).

Click **Add Expression**.



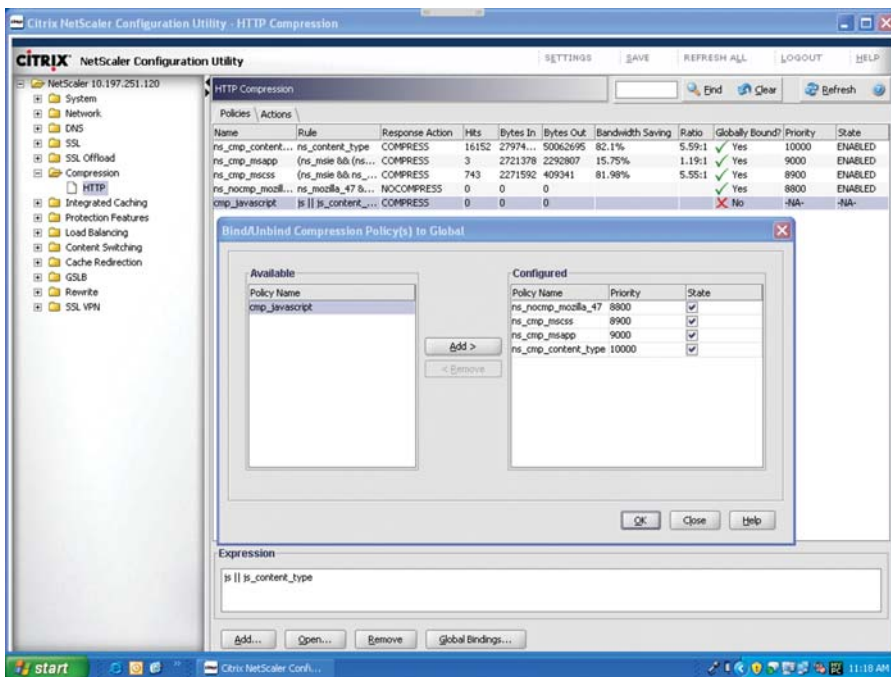
Click **Create** to create the Compression Policy.

The creation of this compression policy will identify and compress the http/https request associated with the javascript policy.

Once the new compression policy is created, click on the Global Bindings button at the bottom of the **HTTP** sub-node window.

Select the **cmp_javascript** policy and move it from **Available** to **Configured**.

This will bind the new javascript compression policy to the global settings.



CITRIX NETSCALER STATIC CACHING

Standard HTTP caching requires no web application knowledge. In most scenarios it can be turned on transparently. This enables caching of static content. For example, image files are generally static and can be cached by NetScaler. Dynamically generated application content is typically not cacheable using standard caching. You can adjust the standard caching settings to change the maximum cacheable response size, the VIA header string, and make other such minor customizations.

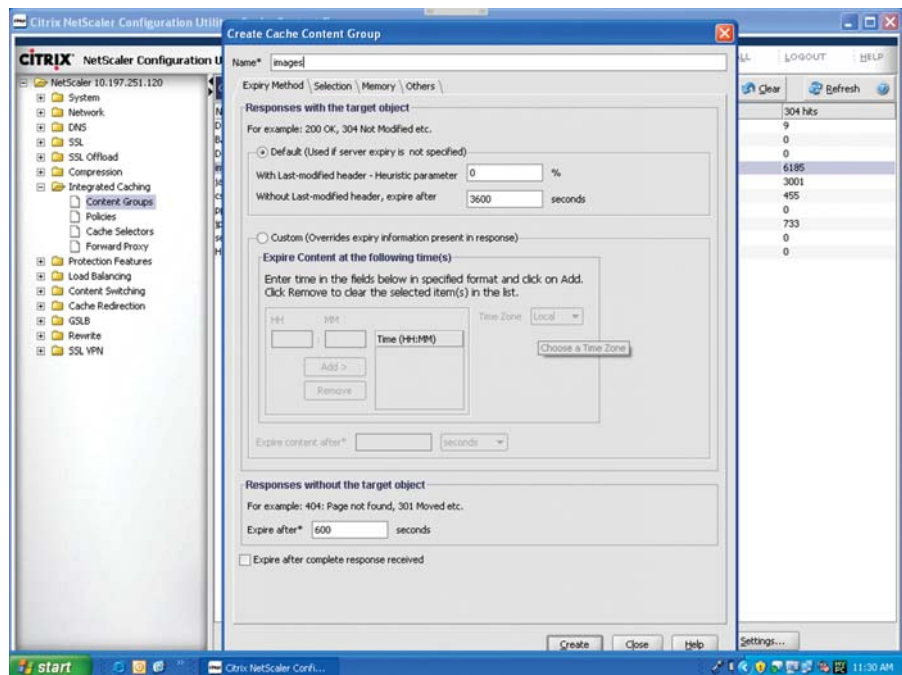
Every object cached by the NetScaler Integrated Cache is made a member of a **Content Group**. The association happens at the time the object is being downloaded and stored. This association is declared in the policy that resulted in the caching of this object.

To configure Static Caching, a **Content Group** must be created.

Go to the Navigation Panel (left side of the main NetScaler Configuration Utility) and expand the **Integrated Caching** node.

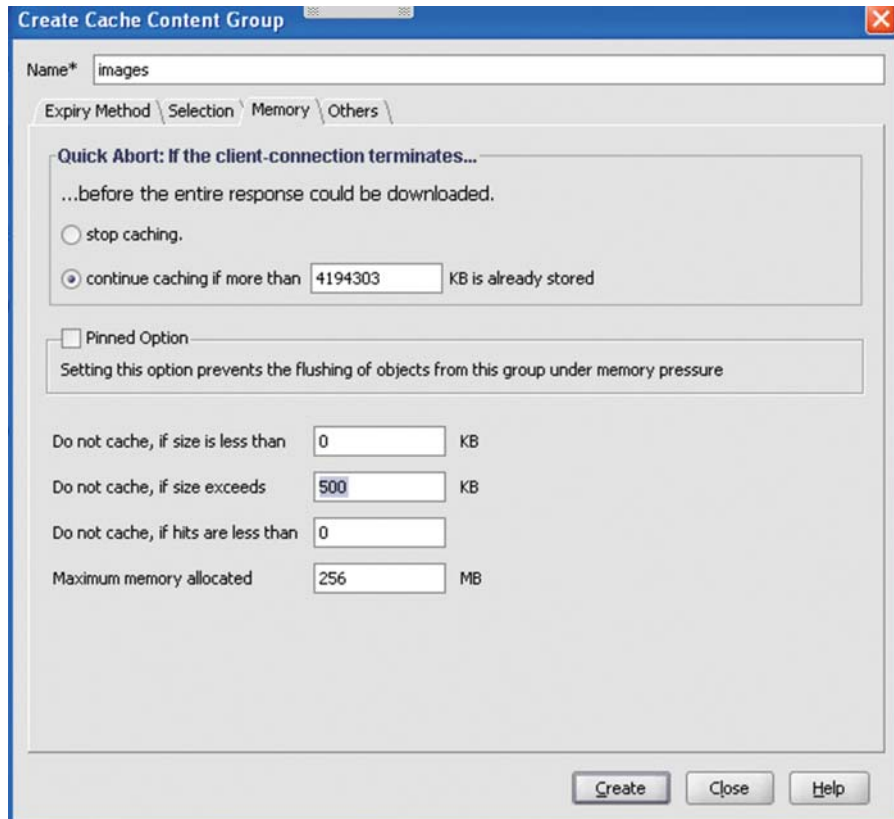
Select the Sub-node **Content Group** and click

In the **Name*** field of the content group, type **images**.



Due to larger than average page sizes generated by the application, the Max Response Size setting for AppCache should be increased from the default of 80 KB to 500 KB.

Select the **Memory** tab to make the change.



The screenshot shows the 'Create Cache Content Group' dialog box with the 'Memory' tab selected. The 'Name*' field contains 'images'. The 'Quick Abort' section has two radio buttons: 'stop caching.' (unselected) and 'continue caching if more than 4194303 KB is already stored' (selected). The 'Pinned Option' checkbox is unchecked. The 'Do not cache' settings are: 'Do not cache, if size is less than 0 KB', 'Do not cache, if size exceeds 500 KB', 'Do not cache, if hits are less than 0', and 'Maximum memory allocated 256 MB'. The 'Create', 'Close', and 'Help' buttons are at the bottom.

Once the Memory settings have been completed, Click **Create** to create the **images** Content Group.

Repeat the steps above to create more Content Groups with the following names:

javascript
css

The creation of these content groups will store all static http/https objects that have matched the static caching policies for images javascript, and css.

To create the Cache Policy for the Content Groups, go to the Navigation Panel (left side of the main NetScaler Configuration Utility) and expand **Integrated Caching** node.

Select the Sub-node **Policies** and click **Add**.

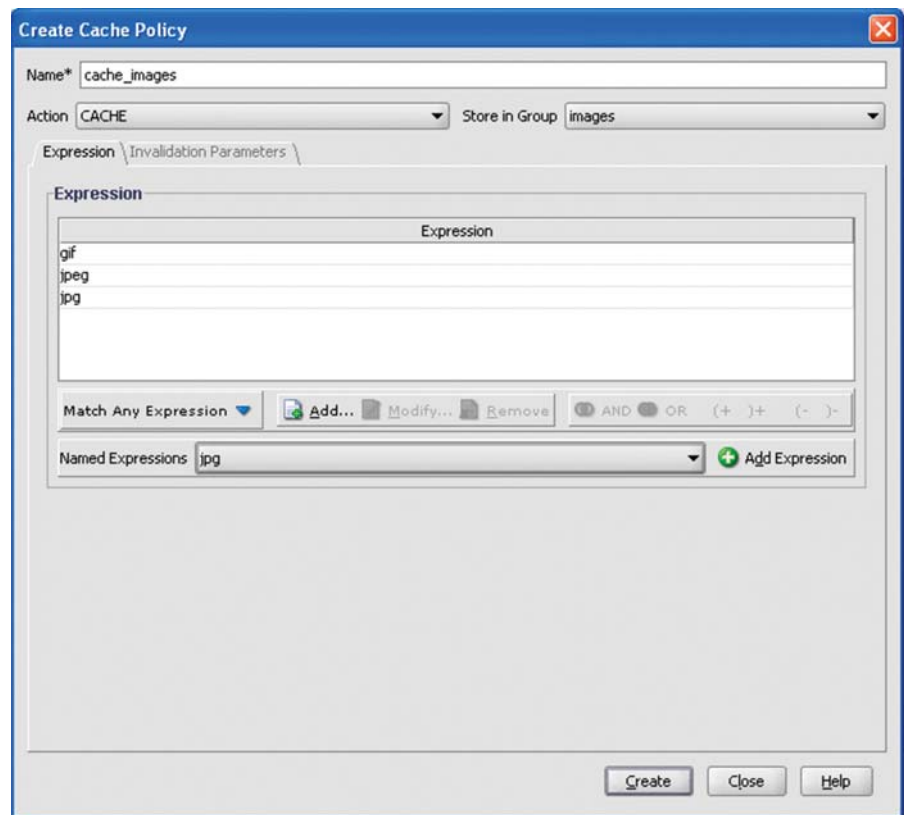
Create Cache policy for **images**:

In the **Name*** field of the Policy, type **cache_images**

In the **Store in Group** pull down box, select **images**.

In the **Named Expressions pull down box**, select **gif** (Policy Expression created in the **System** Node) and click **Add Expression**. Repeat this step to include the Named Expressions **jpeg** and **jpg**.

Click **Create** to create the Cache policy.



The creation of this caching policy will cache all of the static http/https objects that match the gif, jpeg, and jpg expressions and store them in the images content group.

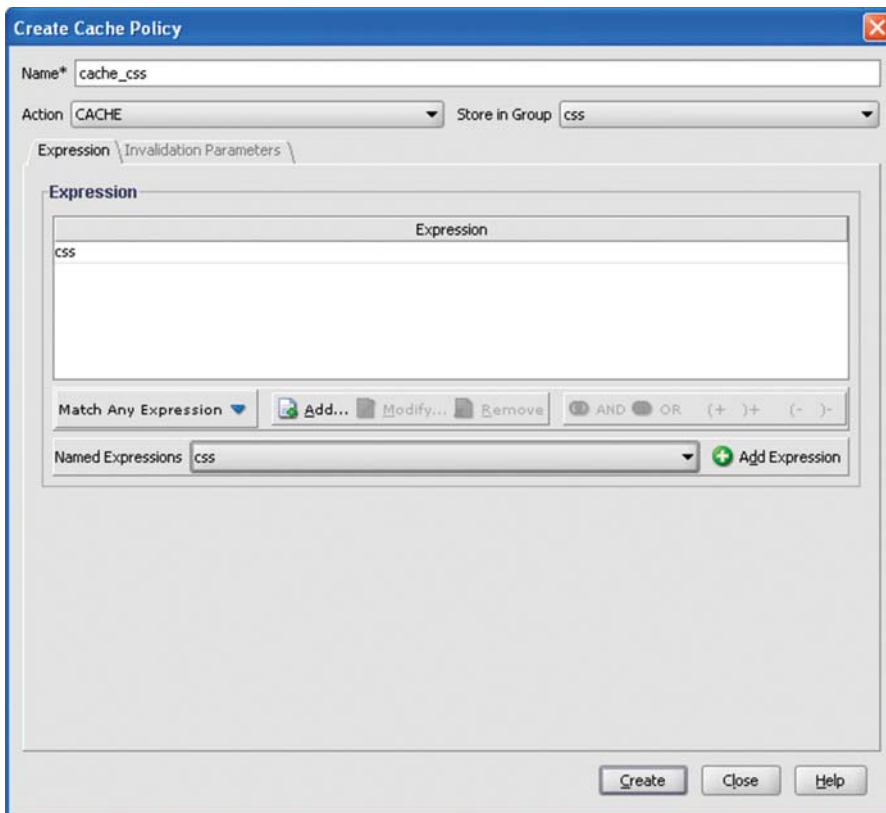
Create Cache policy for **css**:

In the **Name*** field of the Policy, type **cache_css**

In the **Store in Group** pull down box, select **css**.

In the **Named Expressions pull down box, select css** (Policy Expression created in the **System Node**) and click **Add Expression**.

Click **Create** to create the Cache policy.



The creation of this caching policy will cache all of the static http/https objects that match the expression for cascading style sheets and store them in the css content group.

Create Cache policies for **javascript**:

In the **Name*** field of the Policy, type **cache_js_content_type**

In the **Store in Group** pull down box, select **javascript**.

In the **Named Expressions** pull down box, select **js_content_type** (Policy Expression created in the **System** Node) and click **Add Expression**.

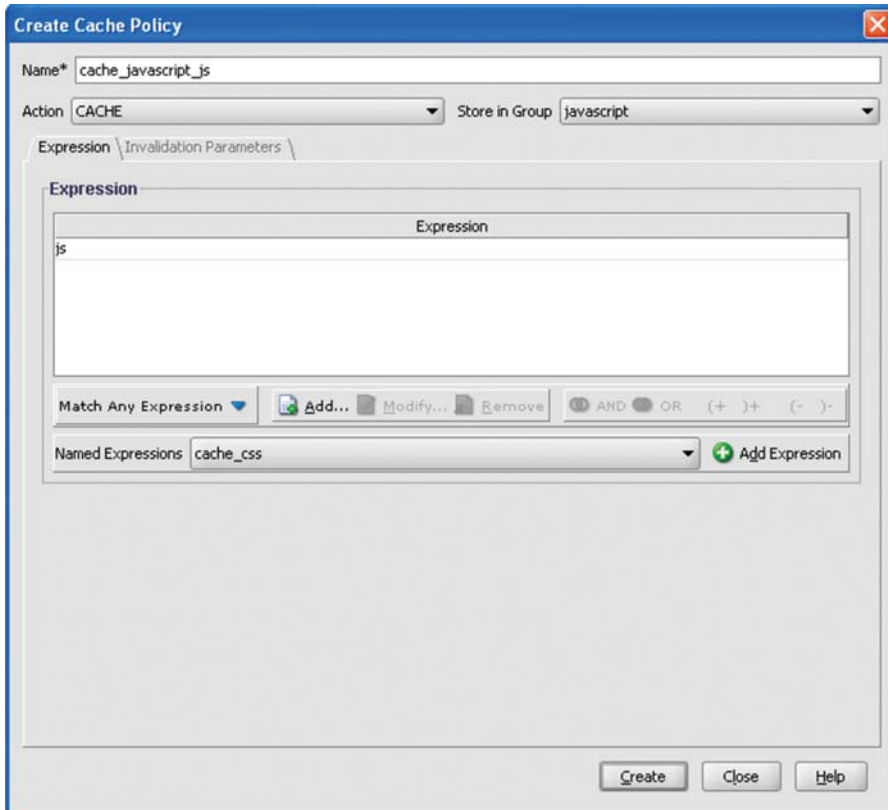
Click **Create** to create the Cache policy.

The screenshot shows the 'Create Cache Policy' dialog box with the following configuration:

- Name*:** cache_js_content_type
- Action:** CACHE
- Store in Group:** javascript
- Expression:** js_content_type
- Named Expressions:** cache_css

Buttons at the bottom right: Create, Close, Help.

In the **Name*** field of the Policy, **type cache_js**
In the **Store in Group** pull down box, select **javascript**.
In the **Named Expressions** pull down box, select **js** (Policy Expression created in the **System** Node) and click **Add Expression**.
Click **Create** to create the Cache policy.



CITRIX NETSCALER LOAD BALANCING

Perform the following steps to configure NetScaler Load Balancing with Siebel Server 2007

Server Monitoring

Prior to configuring Load Balancing, the parameters for Server health monitoring should be considered. The NetScaler Server monitor periodically checks the health of the server by probing a specified destination and taking the appropriate action based on the server response. The NetScaler system has a default TCP based monitor that is automatically bound to each Siebel Service created for load balancing.

Based on the default parameters of the Siebel server, the NetScaler's default TCP monitor will satisfy a basic a basic server health check and will mark the Siebel Service as "Up." The default parameters will be sufficient for a typical Windows based Siebel deployment. For further discussion on customizing other TCP and HTTP based Server monitoring, refer to the Section 6.5 of the NetScaler ICG Volume 1.

The Monitors for use with Siebel are:

- TCP
- HTTP

For further discussion on customizing other TCP and HTTP based Server monitoring, refer to the Section 6.5 of the NetScaler ICG Volume 1

The example below describes the steps in creating a TCP monitor for a Siebel Web **Service**.

Go to the Navigation Panel (left side of the main NetScaler Configuration Utility) and expand the **Monitors** node.

Click **Add**.

In the **Name*** field, enter **Siebel_TCP**

In the **Type** pull down box, select **TCP**

The default parameters are:

Interval : 5

Response Timeout: 2

Date Time: 30

Retries: 3

The screenshot shows the 'Create Monitor' dialog box. The 'Name*' field contains 'siebel_tcp_web' and the 'Type' dropdown is set to 'TCP'. The 'Standard Parameters' section includes the following fields and values: Interval (5), Response Timeout (2), Down Time (30), Deviation (empty), Retries (3), Destination IP (.), Destination Port (empty), Dynamic Timeout (empty), Dynamic Interval (empty), Resp Timeout Threshold (empty), and Action (NONE). The 'Special Parameters' section is empty with the text 'No specials parameters to be configured.' At the bottom, there are checkboxes for 'Enabled', 'Reverse', 'Transparent', 'Secure', and 'LRTM (Least Response Time using Monitoring)', and buttons for 'Create', 'Close', and 'Help'.

NOTE: After the monitor is bound to a Service, the system sends periodic requests to the server. By default the probe interval is 5 seconds. The response from the servers must be received not later than configured response timeout. If the configured number of probes fails, the server is marked DOWN and the next probe is sent after the configured down time.

Repeat steps for creating a **Monitor** for the Siebel Application Server.

Create Monitor

Name* Type

Standard Parameters

Interval

Response Timeout

Down Time

Deviation

Retries

Destination IP

Destination Port

Dynamic Timeout

Dynamic Interval

Resp Timeout Threshold

Action

Custom Header

Enabled Reverse

Transparent Secure

LRTM (Least Response Time using Monitoring)

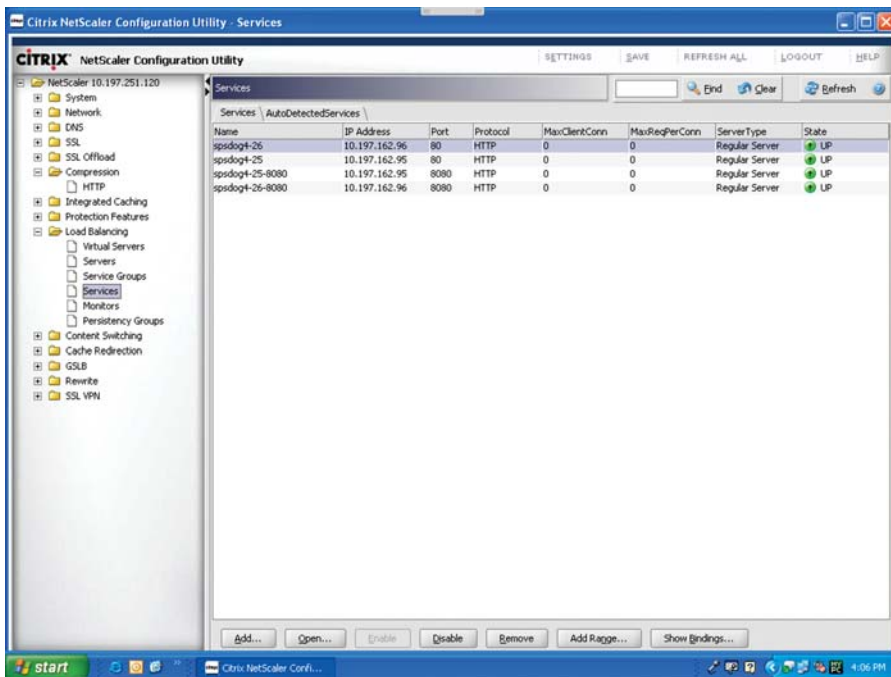
Special Parameters

No specials parameters to be configured.

Server Load Balancing

The first step in configuring Load Balancing requires creating Services for Siebel.

Go to the Navigation Panel (left side of the main NetScaler Configuration Utility) and expand the **Load Balancing** node.



Select the **Services** sub-node and click **Add**.

In the **Service Name*** field, enter the name for this Siebel service. In the example below, the Service Name is **Siebel_app_web1**.

In the **Server** field, enter the IP address of the server. In the example below, the IP address is **172.16.10.236**.

In the **Protocol** Field, select HTTP in the pull-down menu.

In the **Port*** Field, enter **80**

In the **Monitors** tab, select **siebel_tcp_web** and Add to the **Configured** list.

Repeat this step for other **Siebel web servers** to be Load balanced.

These newly created services are now configured to accept http requests through port 80 and are ready to be placed in the Siebel Virtual Server.

Create Service

Service Name* Server

Protocol Port*

Enable Service

Monitors \ Policies \ Advanced \ SSL Settings \

Available

Monitors
udp-ecv
dns
ftp
tcps
https
tcps-ecv
https-ecv
ldns-ping
ldns-tcp
ldns-dns
ver-tcp-ecv-6_5
ver-tcp-ecv-15_5
siebel_tcp_app

Configured

Monitors	Weight	State
siebel_tcp_web	1	<input checked="" type="checkbox"/>

For Load Balancing the Application Servers of Siebel CRM 8, a separate service with a unique TCP Port can be created to facilitate this application.

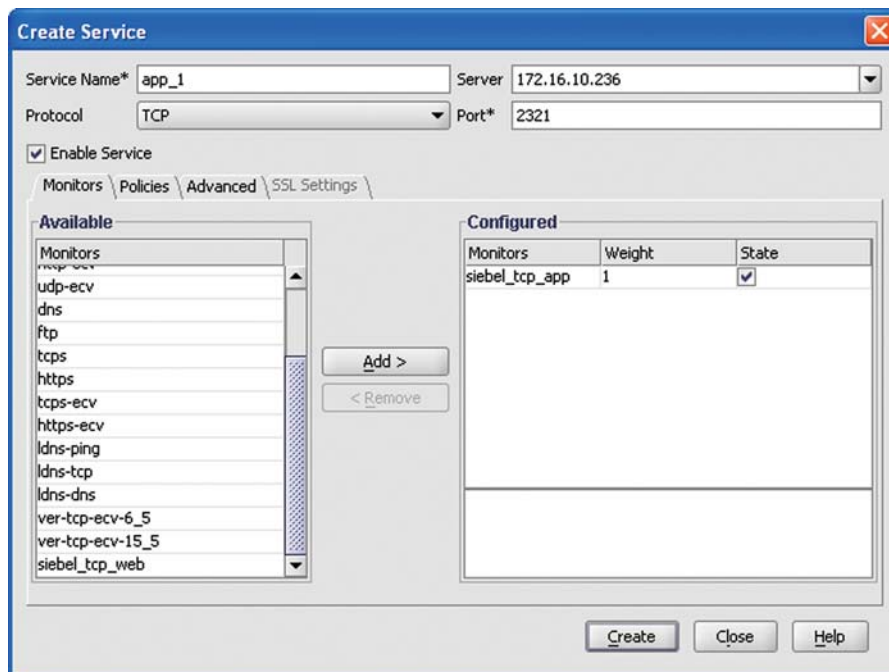
In the **Service Name*** field, enter the name for this Siebel service. In the example below, the Service Name is **app_1**

In the **Server** field, enter the IP address of the server. In the example below, the IP address is **172.16.10.236**.

In the **Protocol** Field, select HTTP in the pull-down menu.

In the **Port Field***, enter **2321**

Repeat these step for other **Siebel Application servers** to be Load balanced.



Once all of the Siebel Services have been configured, a Virtual Server is the next step to be configured for Load Balancing.

Configuring Siebel 8 Web Virtual Server

Go to the Navigation Panel (left side of the main NetScaler Configuration Utility) and expand the **Load Balancing** node.

Select the **Virtual Server** sub-node and click **Add**.

In the **Name*** field, enter the name for this Siebel service. In the example below, the Virtual Server Name is **_http_80**

In the **Server** field, enter the IP address of the server. In the example below, the IP address is **172.16.10.242**

In the **Protocol** Field, select **HTTP** in the pull-down menu.

In the **Port*** Field, enter **80**

To bind the Siebel services to the Virtual Server, Activate the desired **Services** to be load balanced by checking the **Active** boxes.

Creating this Virtual server now enables NetScaler to accept application server requests and load balance them across servers within the Siebel Virtual Server deployment.

Create Virtual Server (Load Balancing)

Name* IP Address* IPv6

Protocol Port*

Network VServer Range Directly Addressable Enable after creating

Services \ Service Groups \ Policies \ Method and Persistence \ Advanced \ SSL Settings \

[Activate All](#) [Deactivate All](#) [Add Service](#)

Active	Service Name	IP Address	Port	Protocol	State	Weight	Dynamic Weight
<input checked="" type="checkbox"/>	Siebel_app_web2	172.16.10.237	80	HTTP	UP	1	
<input checked="" type="checkbox"/>	Siebel_app_web1	172.16.10.236	80	HTTP	UP	1	

Configuring Siebel 8 Application Virtual Server

For load balancing the Siebel Application Servers, select the **Virtual Server** sub-node and click **Add**.

In the **Name*** field, enter the name for this virtual server. In the example below, the Virtual Server Name is **app_vip**.

In the **IP Address*** field, enter the IP address. In the example below, the IP address is **172.16.10.243**.

In the **Protocol** Field, select **TCP** in the pull-down menu.

In the **Port*** Field, enter **2321**

To bind the Siebel services to the Virtual Server, Activate the desired **Services** to be load balanced by checking the **Active** boxes.

The screenshot shows the 'Create Virtual Server (Load Balancing)' dialog box. The 'Name*' field contains 'app_vip', 'IP Address*' is '172.16.10.243', 'Protocol' is 'TCP', and 'Port*' is '2321'. There are checkboxes for 'Network VServer', 'Range', 'Directly Addressable', and 'Enable after creating'. Below is a breadcrumb trail: 'Services \ Service Groups \ Policies \ Method and Persistence \ Advanced \ SSL Settings'. There are links for 'Activate All', 'Deactivate All', and 'Add Service', and a 'Find' button. A table lists services with columns: Active, Service Name, IP Address, Port, Protocol, State, Weight, and Dynamic Weight.

Active	Service Name	IP Address	Port	Protocol	State	Weight	Dynamic Weight
<input checked="" type="checkbox"/>	app_2	172.16.10.237	2321	TCP	UP	1	
<input checked="" type="checkbox"/>	app_1	172.16.10.236	2321	TCP	UP	1	

Buttons at the bottom: Create, Close, Help.

To bind the Siebel services to the Virtual Server, select the **Services** tab and add **Available** services to the **Configured** Services.

Creating this Virtual server now enables NetScaler to accept http requests and load balance them across Siebel Application servers within the Application Virtual Server deployment.

Session Persistence for Siebel

When the NetScaler Load Balancer initially selects a specific Siebel server and directs a client request to this server, all subsequent requests from the same client may need to be sent to the same physical server to access state information for that client.

To enable session persistence for Siebel, NetScaler Cookie based persistence can be configured to insert an HTTP cookie into the client responses. The cookie is inserted into the Cookie header field of the HTTP response. A web browser configured to accept cookies will include it in all subsequent requests to the server.

Open the **Virtual Server** from the **Load Balancing Node**.

Select the **Method and Persistence** tab.

Choose **Cookie Insert** for the **Persistence** type.

Set the timeout to **0**.

NOTE: A value of **0** will not set an expiration time regardless of Cookie version. The expiration time is client software implementation dependent, and usually such cookies expire when the software is properly closed.

The screenshot shows the 'Create Virtual Server (Load Balancing)' dialog box. The 'Name' field contains 'Sharepoint_http_80', 'IP Address*' is '10.197.251.122', and 'Port*' is '80'. The 'Protocol' is set to 'HTTP'. There are checkboxes for 'Network VServer', 'Range', 'Directly Addressable', and 'Enable after creating'. The 'Method and Persistence' tab is selected, showing 'LB Method' as 'Least Connection' and 'Current: Round Robin'. A table lists two servers: 'spsdog4-26' and 'spsdog4-25', both with weight 1, IP addresses 10.197.162.96 and 10.197.162.95, and port 80. The 'Persistence' section is set to 'COOKIEINSERT', 'Timeout' is '0 (min)', 'Backup Persistence' is 'NONE', and 'Netmask' is empty. Buttons for 'Create', 'Close', and 'Help' are at the bottom.

Weight	Name	IP Address	Port
1	spsdog4-26	10.197.162.96	80
1	spsdog4-25	10.197.162.95	80

Repeat this process for all Siebel Virtual Servers.

CONCLUSION

With the latest release of the Siebel CRM suite, Oracle is allowing users to work with a wide ranging array of customer resource applications in a collaborative manner and increasing productivity across the enterprise. But with the trends towards outsourcing and geographically dispersed workforces, sales channels, and partners, the deployment of a centralized customer resource management tool is proving to be more and more challenging.

To ensure the optimal user experience with Siebel CRM, deployment with Citrix NetScaler is advised. The installation demonstrated in this guide includes the compression, caching and load balancing features of NetScaler. These capabilities help overcome the protocol inefficiencies and distance limitations of running applications over a wide area network.

This guide highlights the simplified nature of NetScaler's AppExpert Visual Policy Builder to easily create in a user friendly GUI interface a series of powerful policy expressions and then to apply them to the various functional modules. No need for TCL script writing or code generation is required. Additional modules including web application firewall, SSL access gateway, IPV6 to IPV4 translation, and EdgeSight can be similarly incorporated in a straightforward manner. Contact Citrix technical support for any installation and configuration questions you may have.



ABOUT CITRIX:

Citrix Systems, Inc. (Nasdaq:CTXS) is the global leader and the most trusted name in application delivery infrastructure. More than 180,000 organizations worldwide rely on Citrix to deliver any application to users anywhere with the best performance, highest security and lowest cost. Citrix customers include 100% of the Fortune 100 companies and 98% of the Fortune Global 500, as well as hundreds of thousands of small businesses and prosumers. Citrix has approximately 6,200 channel and alliance partners in more than 100 countries. Annual revenue in 2006 was \$1.1 billion. Learn more at www.citrix.com.