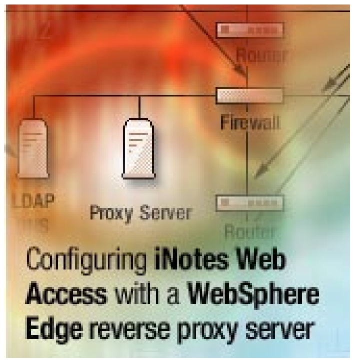
Lotus Developer Domain
The Technical Resource for Lotus software
LDD Today



Configuring iNotes Web Access with a WebSphere Edge reverse proxy server

Level: Advanced
Works with: iNotes Web Access
Updated: 03-Mar-2003

by David Byrd
and Tim Speed

You finally take that once-in-a-lifetime vacation. Walking around the streets of Tokyo, you marvel at the fact that you have left your pager, PDA, laptop, and cellphone back at home. (In fact, your spouse threatened divorce if the laptop went on the trip). You are having the time of your life, with no interruptions or worries—until you happen to pass by a widget shop and suddenly remember: You forgot to place that order for 100,000 widgets before you left home! You try to use a local phone, but no luck—you don't have a calling card and can't figure out how to use the phone. Also, at this time of day in Japan, your home office is closed. In a mounting panic, you consider flying home to place the order in person. But then you spy an Internet café. Salvation! You tell your spouse you'll return in 10 minutes and rush into the café.

With tokens in hand, you type in a URL. The browser displays a login screen, and you enter your user name and password. With surgical precision, you quickly type a memo authorizing an order of 100,000 widgets. You submit the order, check for any important emails while you're at it, then it's off to meet your spouse for lunch. Vacation saved, and you kept your business from losing a large order. Truly a success story!

What magic allowed you do to this securely from a random Internet café? Fortunately, your company uses Domino and iNotes Web Access, enterprise-ready messaging solutions from IBM and Lotus. This provides robust, Notes client calibre messaging features via a Web browser, including email, calendar & scheduling, PIM, task management, and personal journaling. And by using Domino Off-Line Services (DOLS) you can even use these features off-line.

All well and good you say, but what about security? How can we make our servers available to Web users, yet still ensure our data is safe from unauthorized access? As you may have already guessed, we'll be talking about that very subject in this article. In our Tokyo example, we assume you're an Internet roaming user, using iNotes Web Access to conduct secure business through a public browser. To do this, you need an Internet DNS name for public CA-supported Secure Socket Layer (SSL) access to your email. This article explains how to set up your environment to do this, using a WebSphere Edge proxy server to ensure your corporate intranet remains secure from unauthorized users.

We assume that you're an experienced system administrator familiar with Domino and Edge server environments.

A quick review of security

Thanks to some really smart people at the **World Wide Web Consortium** we have a set of standards documents (called Request for Comments or RFCs) that provide rules for how a browser connects to a Web server. The RFCs of particular interest to this article are:

- RCF1738 (URL Definitions)
- RFC1521 (HTTP 1.1)
- RFC2251 (LDAP)

Domino uses these standards to provide Web access to servers. Of course, to do this securely, we need a way to identify and authenticate the person trying to access your server, especially when sensitive data, such as your mail file, is involved. So Domino provides several types of user authentication, including:

- Basic (user name/password) with or without SSL
- Session-based (with or without SSL)
- User certificate-based
- LDAP authentication-based via Directory Assistance
- LTPA (Lightweight Third Party) -based

(The first two authentication methods in the preceding list—basic and session-based—are briefly described later in this article. For more information on other types of authentication, see the **Domino Administrator 6 help**.)

Domino, using one of these authentication methods, asks requestors to identify themselves before allowing access. In the real world, we ask people for their names. In the computer world, we frequently do the same thing—we ask for a name or a user ID. Requiring users to provide both an ID and a password further increases security.

Basic authentication
The simplest form of authentication is called basic authentication. This method only requires the user to supply a valid name and password to gain access. For example, using a browser, you attempt to access your email. When you do, the Domino server returns a status code back to the browser (example – 401). This status code tells the browser to generate a prompt for your user name and password. This data is sent back to the server. In basic HTTP authentication, the password is passed over the network unencrypted—not as plain text, but as "uuencoded" text.

Hackers can use IP sniffers and scanners to capture copies of all packets that pass between a client and server during a session. This information is then available in an unencrypted, plain text format. Anyone watching packet traffic on the network now sees the uuencoded password "in the clear," but the password can be easily decoded by anyone who catches the right network packet. (The uuencode process is not encryption, anyone can download a program to decode the packet and get your password.) In fact, most authentication options that involve a user name and password pass the credential information as plain text—in other words, the bad guys can intercept your user name as well as your password.

Session-based (SSL) authentication
So how do you keep your user name, password, and other data safe? The answer is SSL (Secure Socket Layer). Recall our opening Tokyo story. Using SSL, the transaction from the Internet café can be encrypted, so you know your user name and password (and all other data) are safe.

One of the most basic functions of SSL is message privacy. SSL can encrypt a session between a client and a server so applications can exchange and authenticate user names and passwords without exposing them to eavesdroppers. With SSL, all transmissions following the initial handshake are encrypted to prevent transmissions from being captured. The client and server prove their identities by exchanging certificates. All subsequent traffic between the SSL server and SSL client is encrypted using a key and an encryption algorithm negotiated during the SSL handshake, which occurs at session initialization. Next, the SSL protocol ensures that messages between the sender and receiver have not been tampered with during transmission. This ensures a secure channel between the client and server.

SSL uses a combination of mathematical functions known as hash functions. Included in this process is server authentication. This is the process of determining the server identity via the exchange of certificates. A server's identity is coded into a public-key certificate that is exchanged during the SSL handshake.

SSL was designed to make its security and services transparent to end users. Normally a user follows a URL (see **RFC1738**) to a page that connects to an SSL-enabled server. By default, the SSL-enabled server accepts connect requests on port 443. (The default port for non-SSL access is port 80.) When the handshake process connects to port 443, this establishes the SSL session. All traffic between the server and client is now encrypted—meaning user names and passwords are now secure.

Authorization and trusted networks
Authorization is the process in which the user is granted an appropriate level of access to an object. In the case of a mail file, we can control authorization through standard Domino functionality. All we need to do is add a user name (or Group) into the ACL of the mail file. The native Domino registration process manages this for us.

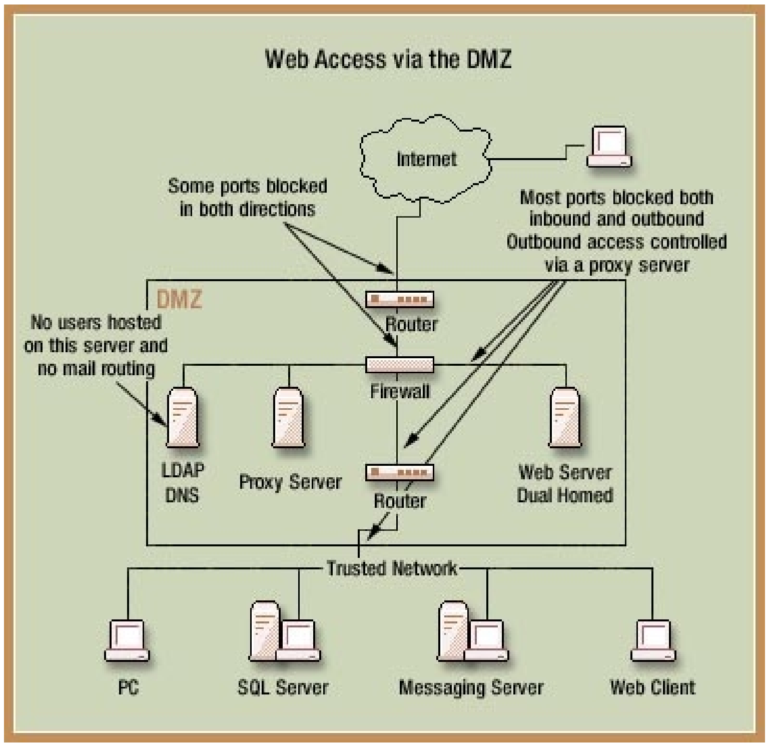
However, once you have decided to expose any part of your computing infrastructure to the Internet, you must define which areas of the networks you use are *trusted* and *untrusted.*In most situations, the Internet is considered an untrusted network. As a rule of thumb, once your data is on the Internet, it is there for all to see.

A trusted network is the network you use to conduct internal business. The trusted network typically supports the back-end systems, internal-only Web pages, data processing, messaging, and in some cases, internal instant messaging. In many companies, the trusted network allows interaction between systems directly without encryption. Also various protocols can exist within the trusted network without any type of filtering or even virus scanning.

As a result, a trusted network is not always a secure network. In fact, the trusted network often cannot be trusted. For example, an internal network can consist of many different networks. These include new acquisitions, joint ventures, international access points, and multiple connections to the outside Internet world.

Demilitarized zone (DMZ)
To help ensure your trusted network is indeed trustworthy, you can create a single access point to the outside world via a DMZ (demilitarized zone). A DMZ is an isolated network placed as a buffer between a company's trusted network and the untrusted network. The DMZ prevents outside users from gaining direct access to the trusted network.

The following diagram shows a basic setup for a DMZ. The untrusted network (in this case the Internet) is shown on the top of the DMZ. The trusted network is shown at the bottom of the diagram. To access the trusted network, you need to cross the DMZ.


Web Access via the DMZ
Internet
Some ports blocked in both directions
Most ports blocked both inbound and outbound Outbound access controlled via a proxy server
DMZ
No users hosted on this server and no mail routing
Router
Firewall
LDAP DNS
Proxy Server
Router
Web Server Dual Homed
Trusted Network
PC
SQL Server
Messaging Server
Web Client

Most DMZs are configured through a set of rules. These rules are controlled by policies and implemented as a set of procedures for your organization. One of the most common rules states a single protocol cannot transverse the DMZ. Another common rule is that you cannot execute a TCP/IP transaction directly into the trusted network. In other words, every TCP/IP transaction must be terminated in the DMZ and re-established before continuing into the trusted network. This is what the DMZ does: It keeps untrusted traffic from entering the trusted network. It is the job of the DMZ to control and limit access to the trusted network by filtering, authentication, and in some cases, completely blocking traffic as needed.

So what can a DMZ do for inbound traffic? The following is a partial list:
- Filter and manage against various attacks such as denial of service (DoS) or TCP/IP sync floods.
- Scan email messages for viruses, content, and size.

- Block passive eavesdropping/packet sniffing.
- Block application-layer attacks.
- Block port scans.
- Limit access to the trusted network via a single protocol.
- Block IP address spoofing.

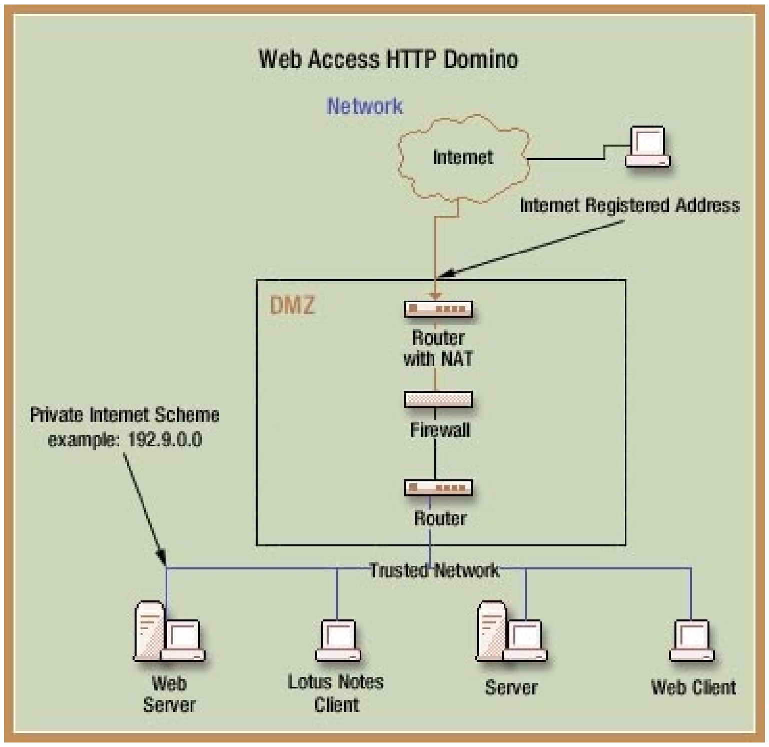
For example, some companies make a partial copy of the data in the trusted network, then move (or replicate) it to servers in the DMZ. This can be a good idea, unless you have a strict policy stating no business data is to be placed into the DMZ. In this case, you may need an application to intercept requests, authenticate users, and forward the requests into servers in the trusted network. So as you can see, one size does not fit all.

Also consider the case of Domino: Let's say you have 10 mail files that you want to make accessible through the Internet. You could easily build a small server, place it into the DMZ, and replicate the 10 production mail files. All will work fine and many customers do this. But perhaps you have 2,000 mail files or even 20,000. Now you need to build many servers (and possibly multiple Notes/Domino domains) and manage a large replication schedule. This can quickly become a very significant chore.

So far, we have discussed using the DMZ to control inbound traffic. However, the DMZ is also used to control outbound traffic. In addition, it can hide (mask) the design and configuration of the trusted network. The DMZ can be designed to limit access to the Internet via proxy servers (more on this subject later) and filter servers. These servers (as regulated by the limits set in the policy document) can do the following:

- Control email messages based on destination, size, and content.
- Scan for viruses going out of the DMZ.
- Limit access to unauthorized sites.
- Monitor access to unauthorized sites.

Private networks
One common practice that companies use today is to set up a private network by hiding the internal IP addressing scheme from the outside Internet. Using the routers in the DMZ can help accomplish this. The technique is known as network address translation (NAT). (See **RFC1631** for more information.) The following diagram shows an example where a single address is mapped to a 192.9.0.0 network:

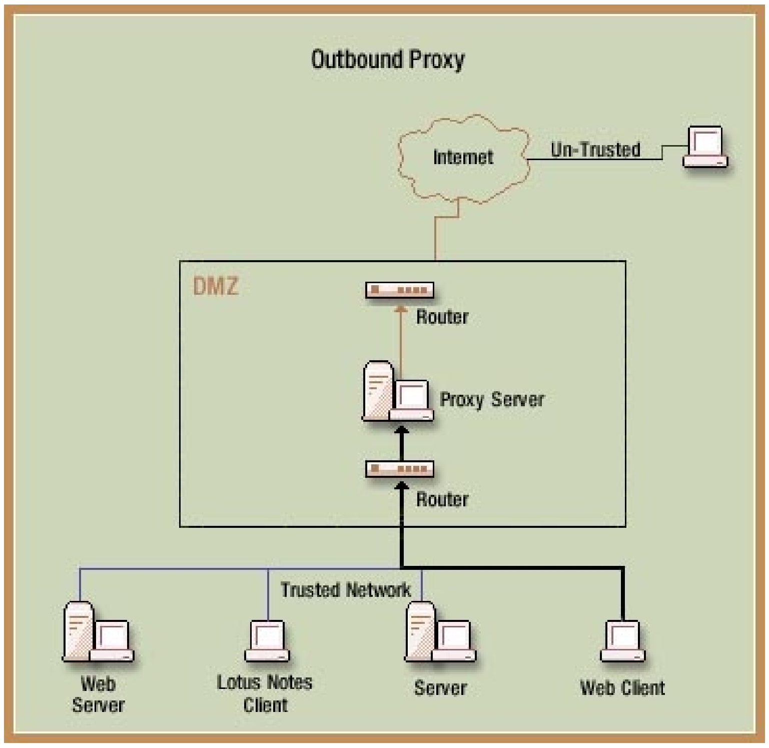


This is very easy and simple to set up. In fact, you can purchase home routers for under $100 to do the same for your cable modem or your DSL connection. Check out **RFC1591** for more information about private address networks. There are many advantages to using private networks:

- Only a very small set of Internet address are needed for all traffic.
- A private address scheme makes it harder for a hacker to attack your network.
- You can block all inbound traffic on the same network. For example, if you are using 192.9.200.0 then anyone sending a packet from that network is a hacker; you instruct your routers to discard these packets.
- You can manage and control Internal routing at will; any changes you make to your routers will not impact routing on the Internet.

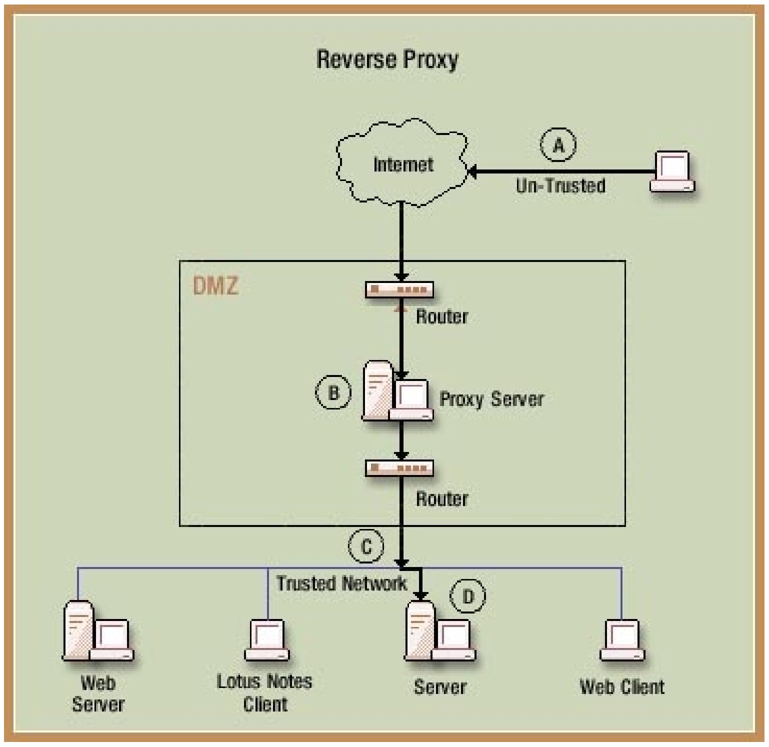
Proxy servers

Another method to control access in and out of your trusted network is through a proxy server. A *proxy server* (also known as an application gateway) is an application that mediates traffic between a trusted network and an untrusted network. This does not remove the need for a firewall to manage the traffic at an IP level, but provides for an application-level firewall. Many proxy servers contain extra logging or support for user authentication. This illustration shows a simple example of an outbound proxy server:



All traffic is scanned before leaving the trusted network. Also all return traffic is scanned before it is returned to the user. Most proxy servers can also provide a caching function. If a page is accessed more than once, it can be cached in the proxy server for faster access.

A related concept is the *reverse proxy.* This is a server located between the Internet and your Web servers, normally in a DMZ. The reverse proxy server intercepts user requests arriving from the Internet and forwards them to the appropriate Web server. (Other applications, such as Sametime, can also be reverse proxied.) The following diagram shows an example of a reverse proxy:



As you can see, the preceding diagram shows four basic steps (A through D) in the reverse proxy process:

A. A browser user issues a request to open a Web page. This URL, via a DNS entry, is directed to the DMZ.
B. The request enters the reverse proxy server and is directed to a destination in the trusted network. This is controlled by a set of rules or configuration documents. For example: If the URL is /mail, then direct traffic to http:// 192.9.200.55/mail
C. The request exits the DMZ and is directed to the target server.
D. The request enters and is processed by the target server.

With the reverse proxy, you can limit access to a single server on your trusted network. The reverse proxy blocks all traffic except for what is allowed in its rule set. This means the reverse proxy stops attempts to access other servers and their resources on the trusted network. “But the bad hackers could still access the server by breaking into the reverse proxy server!” some may exclaim. Not to worry—most reverse proxy servers offer authentication. You can limit access to the DMZ by prompting users with a user name and password at the reverse proxy server.

Installing a reverse proxy environment

So far we've discussed:

- iNotes Web Access: a robust Web based email system from Lotus IBM
- SSL: a network layer security encryption used to protect your user name, password, and data in transit on the Internet
- DMZ: a zone that separates your trusted network from the Internet
- Proxy server: a system to restrict access and data to or from your trusted network
- Reverse proxy: a system to control access to servers in the trusted network

The remainder of this article explains how you can put these all together into a single solution. Our goal is to provide secure access to your mail file on a Domino server in the trusted network. We also want to limit access to the trusted network to just the Domino servers. In this example, we use an IBM WebSphere Edge server as our reverse proxy.

In theory, any reverse proxy can be used for this solution. In practice, the following servers/devices have been used on production enterprise sites:

Reverse proxy:

- WebSphere Edge reverse proxy servers
- Tivoli Policy Director

- Netegrity's Secure Reverse Proxy Server
- Sun's iPlanet
- Neoteris Reverse Proxy Web appliance
- Whale E-Gap

SSL appliance:
- Redline SSL accelerators

WebSphere Edge server
The **IBM WebSphere Edge server** provides services to users who access documents stored on an enterprise's server machines and to internal users who access the Internet. The Edge server can access Web content and can provide Internet access efficiently and economically. As the name Edge server suggests, this software usually runs on machines that are close (in a network configuration sense) to the boundary between an enterprise's intranet and the Internet. The Edge server can operate in many different roles:
- Forward proxy
- Reverse proxy
- Cache proxy
- IP dispatcher (used in a multiple server configuration for load balancing)

In this article, we install an Edge server in a reverse proxy configuration with inbound authentication. This allows us to prompt the user twice, once for access to the DMZ and once for access to the Domino server. Note that single sign-on (SSO) is available, although we will not use this feature in our example.

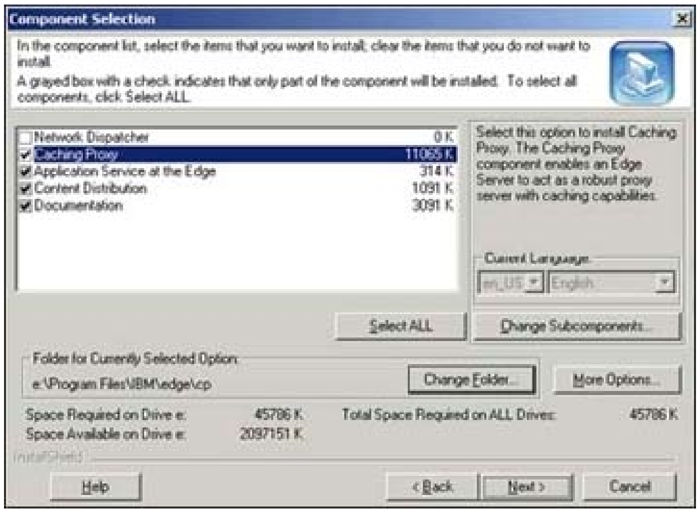
The reason we're prompting users twice is part security and part psychological. Any user who requires access to the trusted network through the Internet needs to access the trusted network via a user name and password. Many companies configure this as a security requirement and make their employees use a different user name and password to access the DMZ. Having two different user names and passwords may add additional security. In any case, you need to determine what works best for your company. Another reason we are showing you two prompts is to describe the basic features of the Edge server.

The Edge server can be installed on these systems:
- AIX
- Solaris
- Linux
- Windows

See the WebSphere Edge Server documentation for hardware and software requirements.

In the following instructions, we assume you have the Edge server software and documentation available.

Installing the Edge server
1. Execute the Edge install package. The install program explodes the files into a temp directory; from there, execute the setup.exe file.
2. Two informational screens appear, followed by the license agreement. After you accept, the Component Selection screen appears:



Component Selection

In the component list, select the items that you want to install; clear the items that you do not want to install.
A grayed box with a check indicates that only part of the component will be installed. To select all components, click Select ALL.

☐	Network Dispatcher	0 K
✓	Caching Proxy	11065 K
✓	Application Service at the Edge	314 K
✓	Content Distribution	1091 K
✓	Documentation	3091 K

Select this option to install Caching Proxy. The Caching Proxy component enables an Edge Server to act as a robust proxy server with caching capabilities.

Current Language: en_US English

Folder for Currently Selected Option: e:\Program Files\IBM\edge\cp

Space Required on Drive e: 45786 K
Space Available on Drive e: 2097151 K
Total Space Required on ALL Drives: 45786 K

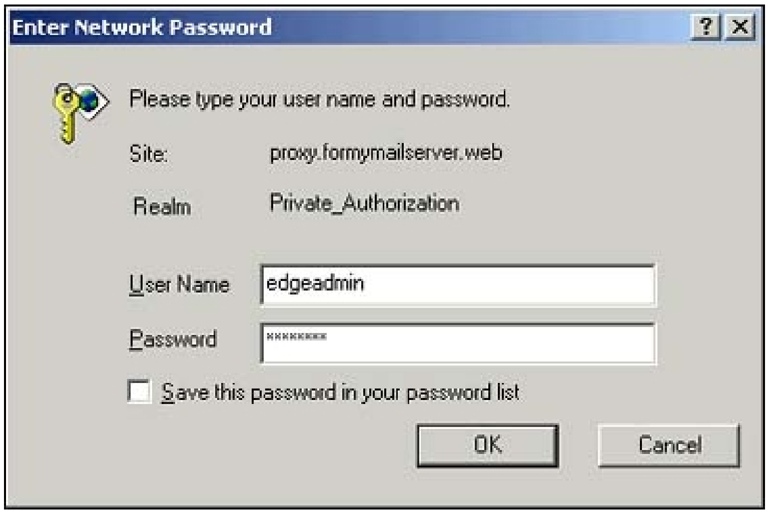
3. This screen is where you select the components you want to enable. Select Caching Proxy, then click Next. This allows you to set up a reverse proxy during the configuration part of the install (described in the next section of this article).
4. The Edge code is installed. Reboot your server, and start the Configuration Wizard to set up the reverse proxy server.
5. At the Select Proxy Behavior prompt, select Reverse Proxy and click Next.
6. At the Select Proxy Port prompt, enter the port number (the default is 80).
7. At the Target Web Server prompt, enter the primary inbound URL.

The Edge server is now installed and ready for configuration. At this point, you may want to review the configuration with the firewall administrator to ensure that all the necessary rule sets have been put in place and are functioning. Specifically, for most configurations only port 443 (SSL) should be allowed in from the Internet into the DMZ address (Proxy.forMyMailServer.Web in this example) for the reverse proxy server. The reverse proxy server should be able to open either a port 80 (non-SSL) or port 443 (SSL) connection back into the internal Domino server (for instance iNotes.forMyMailServer). This type of firewall rule prevents any direct connection from an Internet-based request reaching the internal Domino servers. All traffic must be filtered through the proxy server.

Configuring the Edge server
The configuration of the software itself is performed in two places. The majority of the administration duties for the proxy server are done through a Web browser Administration interface. This allows direct access to 95 percent of the configuration parameters necessary for daily operation of the server. There are some items, however, that are not available through the Administration interface and must be changed by editing the ibmproxy.conf file.

Edge Administration interface
The URL to open the Administration interface is initially http://Proxy.forMyMailServer.Web/admin-bin/webexec /frameset.html. This will change to https once SSL is configured on the system.

Click the URL to open the Administration screen. You are prompted for your user name and password:



Enter Network Password

Please type your user name and password.

Site: proxy.formymailserver.web

Realm Private_Authorization

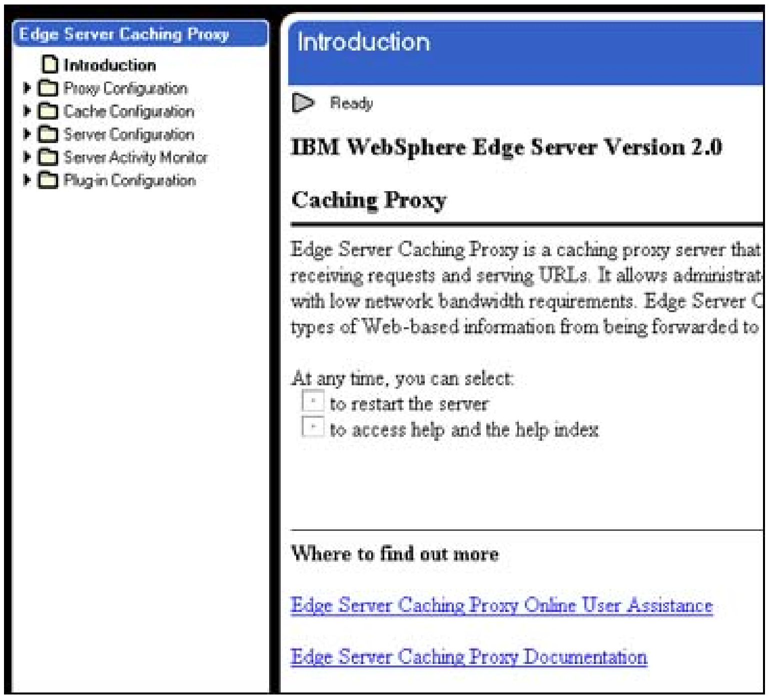
User Name: edgeadmin

Password: ********

☐ Save this password in your password list

(If the Administration screen does not come up, it is likely due to the Configuration wizard putting the Proxy rule first in the rule list under the Mapping rules section. You can fix this issue by manually editing the ibmproxy.conf file located under the c:\program files\ibm\edge\cp\etc\en_US directory. Move the Proxy /* http://iNotes.forMyMailServer.Web/* directive below the # *** ADD NEW MAPPING RULES HERE *** section.)

If all goes well, the Administration interface should appear as shown in the following screen. This interface is the portal into most operational functions of the proxy server. With just a few exceptions, all configuration changes can be made from here:



Introduction

Ready

IBM WebSphere Edge Server Version 2.0

Caching Proxy

Edge Server Caching Proxy is a caching proxy server that
receiving requests and serving URLs. It allows administrat
with low network bandwidth requirements. Edge Server C
types of Web-based information from being forwarded to

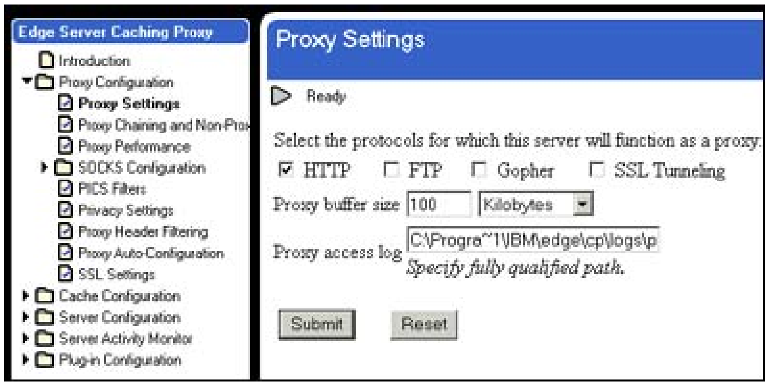
At any time, you can select:
to restart the server
to access help and the help index

Where to find out more

Edge Server Caching Proxy Online User Assistance

Edge Server Caching Proxy Documentation

The Proxy Settings section controls which protocols the proxy server supports. In the following configuration, the server supports only the HTTP protocol; all other protocols should be disabled. The other settings can be left at the default values:



Proxy Settings

Ready

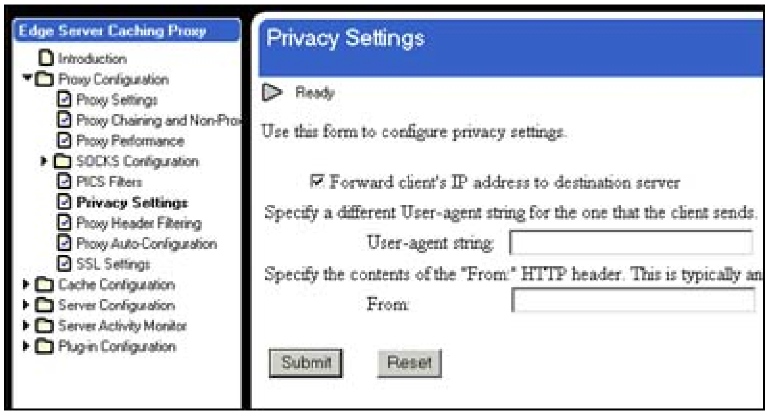
Select the protocols for which this server will function as a proxy:

☑ HTTP ☐ FTP ☐ Gopher ☐ SSL Tunneling

Proxy buffer size 100 Kilobytes

Proxy access log C:\Progra~1\IBM\edge\cp\logs\p
Specify fully qualified path.

The Privacy Setting section allows additional HTTP headers to be passed along with the requests. The only change you should make here is to enable the "Forward client’s IP address to destination server" setting. This adds an additional HTTP header value containing the requesting client’s actual IP address:



Privacy Settings

Ready

Use this form to configure privacy settings.

☑ Forward client's IP address to destination server

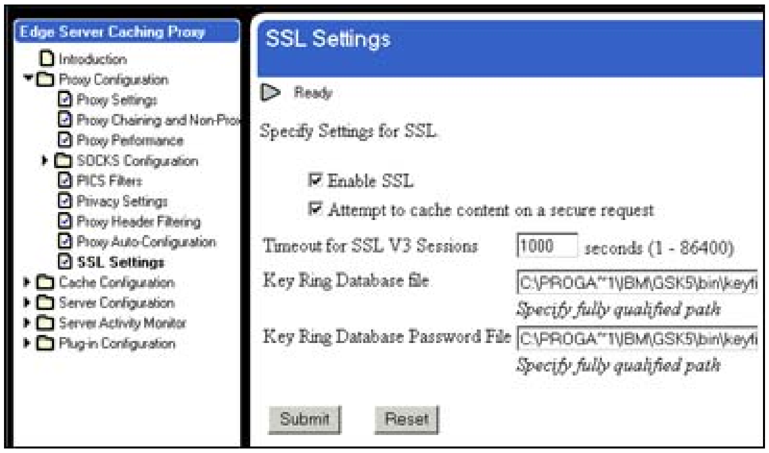
Specify a different User-agent string for the one that the client sends.

User-agent string

Specify the contents of the "From:" HTTP header. This is typically an

From:

The SSL Settings section controls the SSL configuration of the proxy server. This determines how SSL is physically enabled at a server-wide level. For this setting to work, however, you must build and populate the key ring databases with the IBM Key Management tool. (See the Edge Server documentation for more information.)



SSL Settings

Ready

Specify Settings for SSL.

☑ Enable SSL

☑ Attempt to cache content on a secure request

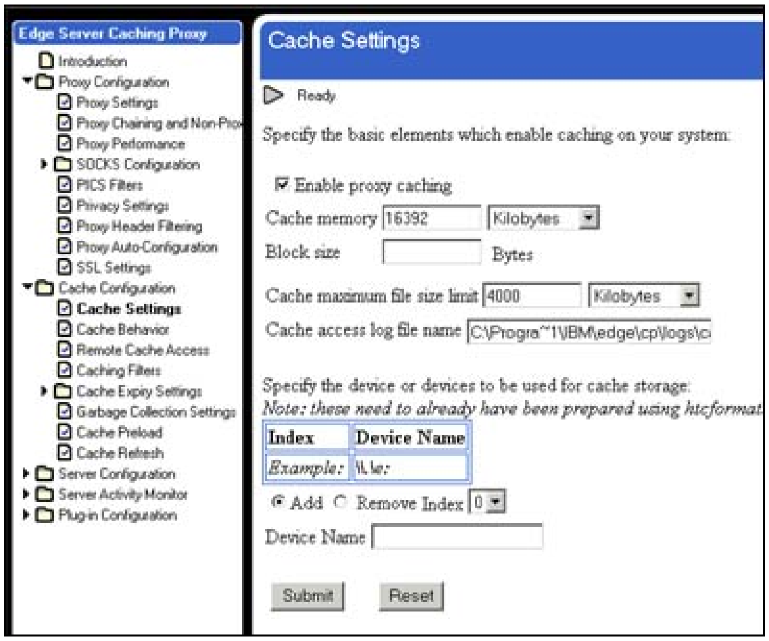
Timeout for SSL V3 Sessions 1000 seconds (1 - 86400)

Key Ring Database file C:\PROGA~1\IBM\GSK5\bin\keyfi
Specify fully qualified path

Key Ring Database Password File C:\PROGA~1\IBM\GSK5\bin\keyfi
Specify fully qualified path

The Cache Settings section controls the caching behavior of the proxy server. The proxy server itself can cache content locally to aid performance. This is very important in most reverse proxy configurations; caching static content can reduce overall load placed on the Domino servers. The main elements cached are images, Java class files, and image resources.

The proxy server has two locations to which it can cache content. The first is within system memory up to the amount specified in the Cache memory field. Memory is the fasting caching device, and this value should be maximized in accordance with the hardware hosting the proxy server. The value of 16392 KB shown in the following screen is the default value. The second cache location is within a dedicated hard drive partition. The location of this caching partition is determined by your physical hardware configuration. In our example, it is not enabled. The steps to enable this function involve the use of the htcformat utility and are outlined in the Edge Server documentation.



Cache Settings

Ready

Specify the basic elements which enable caching on your system:

[✓] Enable proxy caching

Cache memory 16392 Kilobytes

Block size Bytes

Cache maximum file size limit 4000 Kilobytes

Cache access log file name C:\Progra~1\IBM\edge\cp\logs\c

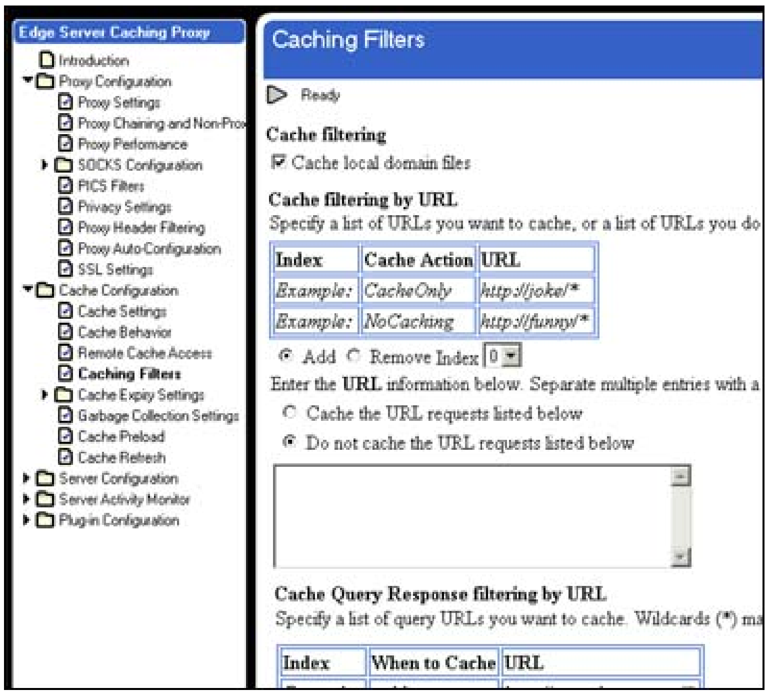
Specify the device or devices to be used for cache storage:
Note: these need to already have been prepared using htcformat

Index	Device Name
Example:	\\.\e:

[✓ Add] Remove Index 0

Device Name

The Caching Filters section allows for the proxy server to cache content retrieved by the reverse proxy. Each respective hosted site must be defined within the "Cache Query Response filtering by URL" field. The URL format is */internal.host.name/*. This represents a wildcard response coming from virtually any reverse proxy request. The caching functions of the proxy server are still bound by the expiration information contained within the HTTP header. This prevents dynamic content from being cached when it should not be.



Caching Filters

Ready

Cache filtering

[✓] Cache local domain files

Cache filtering by URL

Specify a list of URLs you want to cache, or a list of URLs you do

Index	Cache Action	URL
Example:	*CacheOnly*	*http://joke/**
Example:	*NoCaching*	*http://funny/**

[✓ Add] Remove Index 0

Enter the **URL** information below. Separate multiple entries with a

Cache the URL requests listed below

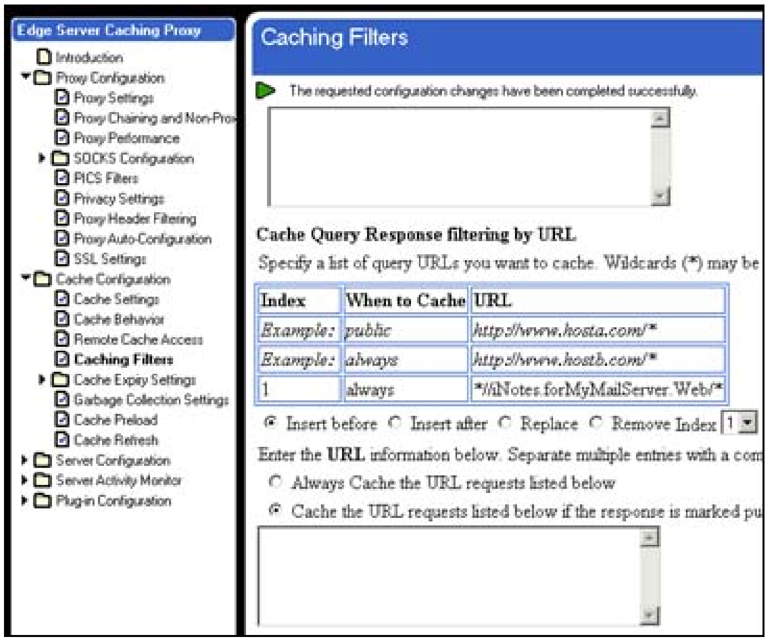
[✓ Do not cache the URL requests listed below]

Cache Query Response filtering by URL

Specify a list of query URLs you want to cache. Wildcards (*) ma

Index	When to Cache	URL

The following screen is the continuation of the Caching Filters form:



Caching Filters

The requested configuration changes have been completed successfully.

Cache Query Response filtering by URL

Specify a list of query URLs you want to cache. Wildcards (*) may be

Index	When to Cache	URL
Example:	*public*	*http://www.hosta.com/**
Example:	*always*	*http://www.hostb.com/**
1	always	*/iNotes.forMyMailServer.Web/*

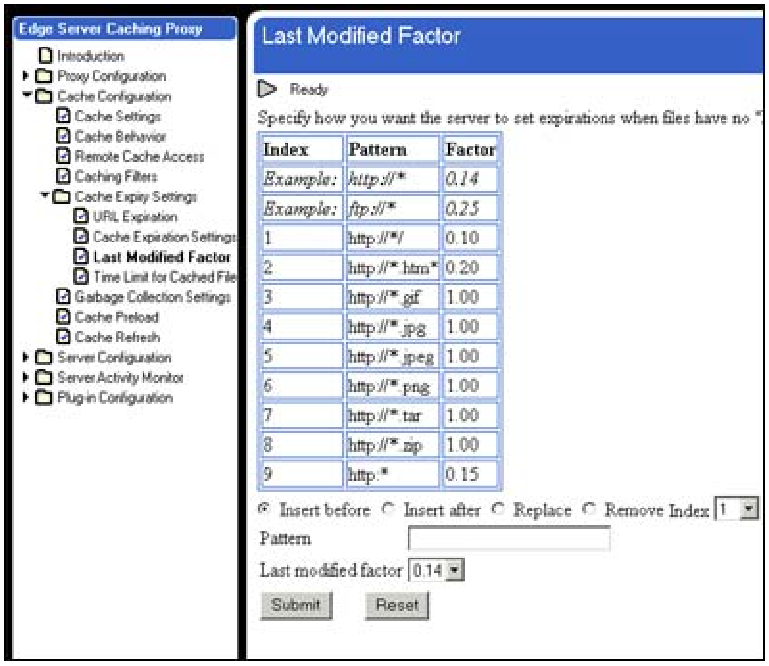
[✓ Insert before] Insert after Replace Remove Index 1

Enter the **URL** information below. Separate multiple entries with a com

Always Cache the URL requests listed below

[✓ Cache the URL requests listed below if the response is marked pu]

The Last Modified Factor section allows more refined control over the expiration time of explicit Domino design elements cached locally on the Edge server. The two main items initially configured are the ?OpenImageResource and ?OpenElement&FieldElemFormat=gif URL requests. These represent images, but due to their nature the proxy server does not by default see them as images (which should be cached longer than just standard HTML).



Last Modified Factor

Ready

Specify how you want the server to set expirations when files have no "

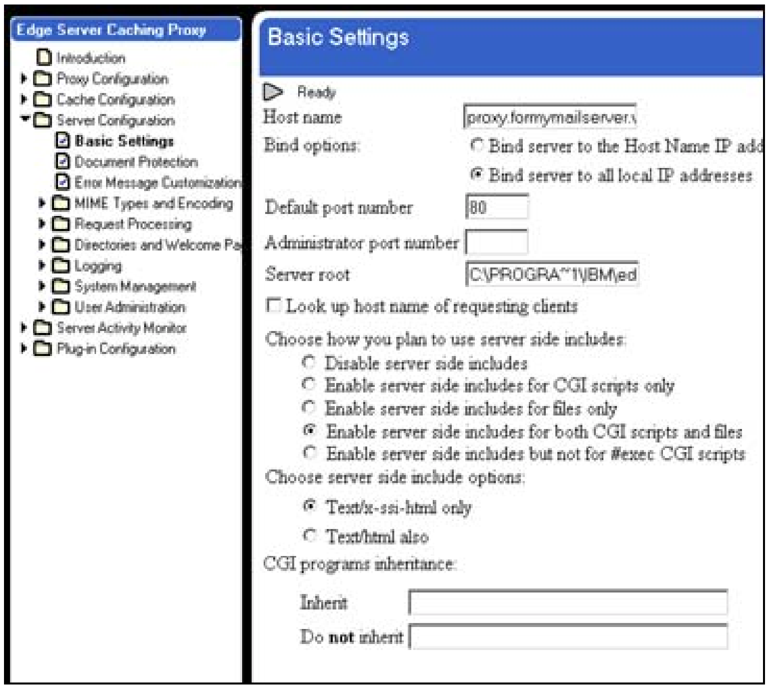
Index	Pattern	Factor
Example:	*http://**	*0.14*
Example:	*ftp://**	*0.25*
1	http://*/	0.10
2	http://*.htm*	0.20
3	http://*.gif	1.00
4	http://*.jpg	1.00
5	http://*.jpeg	1.00
6	http://*.png	1.00
7	http://*.tar	1.00
8	http://*.zip	1.00
9	http:*	0.15

[✓ Insert before] Insert after Replace Remove Index 1

Pattern

Last modified factor 0.14

The Basic Settings section controls the server's host name and IP addresses it listens on. The default port number does not any have effect if the servers are set up to support SSL only (which would likely be the case for extranet-based mail applications). Note that Bind options control which IP addresses are used on the machine. In most cases, this would be bound to all local IP addresses. This section also controls DNS lookups of connecting clients. The option causes the Edge server to resolve each incoming client's IP address with a host name, resulting in processing overhead on the server. Unless there are security or logging reasons for doing so, this option should generally be turned off.



Basic Settings

Ready

Host name: proxy.formymailserver.

Bind options: Bind server to the Host Name IP add [✓ Bind server to all local IP addresses]

Default port number: 80

Administrator port number:

Server root: C:\PROGRA~1\IBM\ed

☐ Look up host name of requesting clients

Choose how you plan to use server side includes:
- Disable server side includes
- Enable server side includes for CGI scripts only
- Enable server side includes for files only
- [✓ Enable server side includes for both CGI scripts and files]
- Enable server side includes but not for #exec CGI scripts

Choose server side include options:
- [✓ Text/x-ssi-html only]
- Text/html also

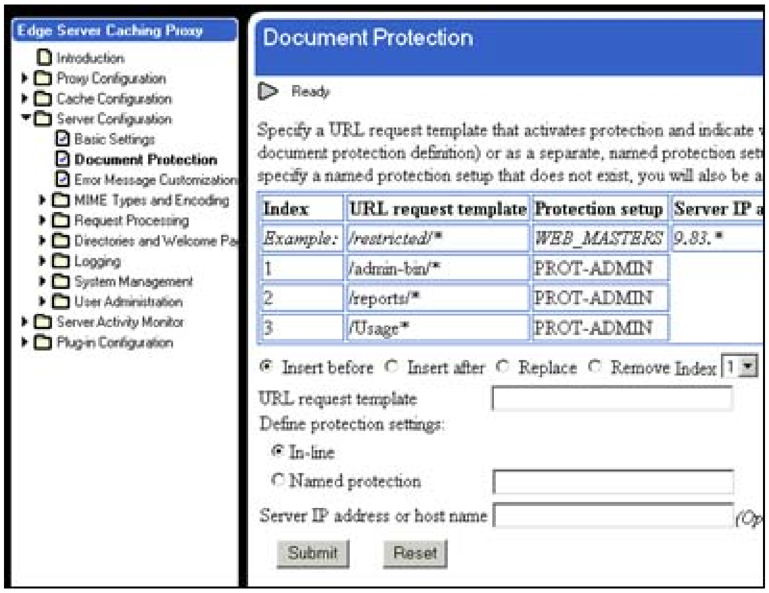
CGI programs inheritance:

Inherit

Do **not** inherit

The Document Protection section allows for special access control to be placed on content served by the Edge server. For instance, the default settings create access control sections for the content under the /admin-bin/*

subdirectories. There are two key elements in this section. The first is that all URL request templates are valid for both Edge local content as well as proxied content. This allows you to place additional restrictions on, for example, a /MyStuff/* folder located on a Domino server. These restrictions could say that only a user called myprotstuff can access the content from an external address of x.x.x.x corresponding to your fixed DSL address. The Edge server requests a name and password before granting access. The credentials come from either a local password store on the Edge server or from, for instance, an LDAP data source. Then after validation by the Edge server the user is routed over to the authorized Domino server for processing.



Document Protection

Ready

Specify a URL request template that activates protection and indicate
document protection definition) or as a separate, named protection set
specify a named protection setup that does not exist, you will also be a

Index	URL request template	Protection setup	Server IP a
Example:	*/restricted/**	*WEB_MASTERS*	*9.83.**
1	/admin-bin/*	PROT-ADMIN	
2	/reports/*	PROT-ADMIN	
3	/Usage*	PROT-ADMIN	

[✓ Insert before] Insert after Replace Remove Index 1

URL request template

Define protection settings:

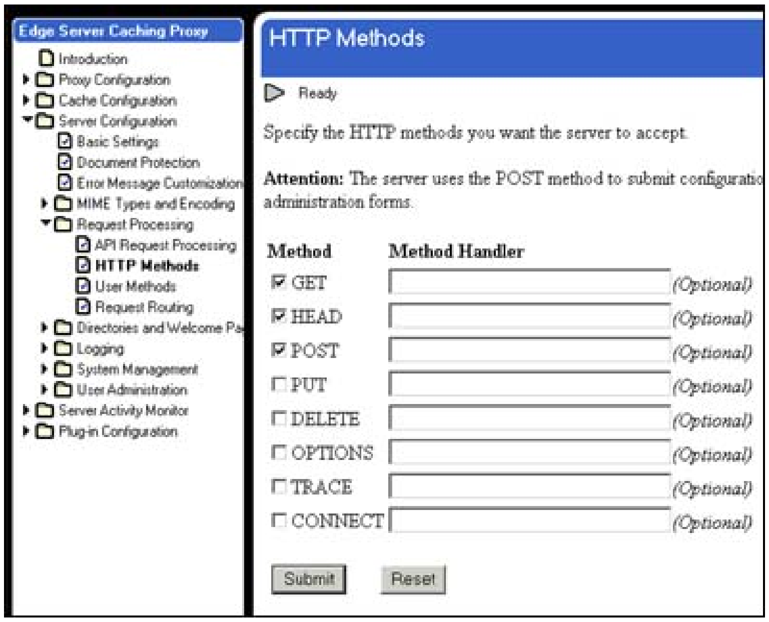
[✓ In-line]

Named protection

Server IP address or host name (Op

Submit Reset

The HTTP Methods section allows you to define request types serviced by the Edge server. There are several turned on by default; the only ones Domino needs to function are GET, HEAD, and POST. The others are unnecessary and could pose a security risk.



HTTP Methods

Ready

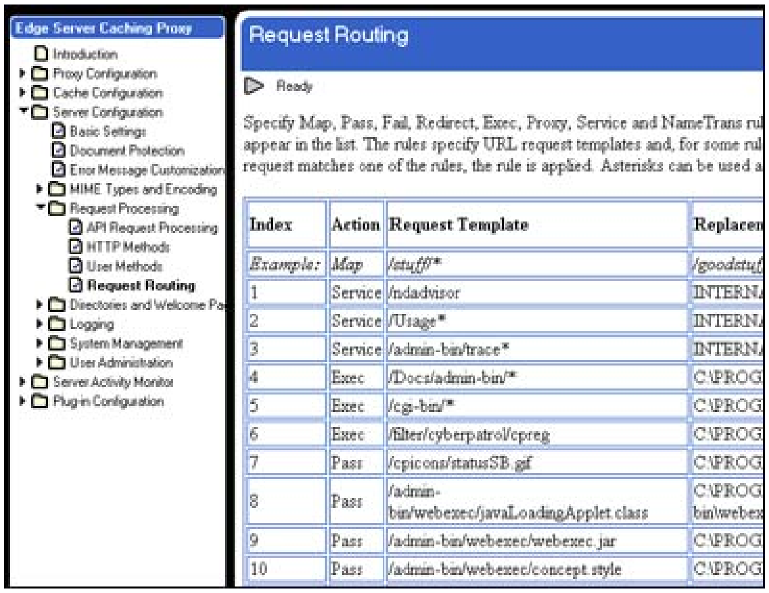
Specify the HTTP methods you want the server to accept.

Attention: The server uses the POST method to submit configuratio
administration forms.

Method	Method Handler	
☑ GET		(Optional)
☑ HEAD		(Optional)
☑ POST		(Optional)
☐ PUT		(Optional)
☐ DELETE		(Optional)
☐ OPTIONS		(Optional)
☐ TRACE		(Optional)
☐ CONNECT		(Optional)

Submit Reset

The Request Routing section is where all the magic happens. These rule sets control the destiny of all incoming requests that the proxy server sees. Therefore, this configuration is vital to the server’s successful operation and performance. The Edge server runs each incoming request through this rule list in the index order, trying to match the Request template and IP address (if defined). This matching is case-sensitive (something non-UNIX Domino sites should bear in mind). This means the Edge server treats /Mail differently from /mail in the request template. Once a request is matched to a rule, the specified action is performed. These actions allow for content to be dropped, passed directly to a local file Edge server resource, executed in the case of CGI application, or proxied to a different location.



Request Routing

Specify Map, Pass, Fail, Redirect, Exec, Proxy, Service and NameTrans rul
appear in the list. The rules specify URL request templates and, for some rul
request matches one of the rules, the rule is applied. Asterisks can be used a

Index	Action	Request Template	Replacen
Example:	*Map*	*/stuff/**	*/goodstuf*
1	Service	/ndadvisor	INTERN
2	Service	/Usage*	INTERN
3	Service	/admin-bin/trace*	INTERN
4	Exec	/Docs/admin-bin/*	C:\PROG
5	Exec	/cgi-bin/*	C:\PROG
6	Exec	/filter/cyberpatrol/cpreg	C:\PROG
7	Pass	/cpicons/statusSB.gif	C:\PROG
8	Pass	/admin-bin/webexec/javaLoadingApplet.class	C:\PROG bin\webex
9	Pass	/admin-bin/webexec/webexec.jar	C:\PROG
10	Pass	/admin-bin/webexec/concept.style	C:\PROG

The proxy rule for a "pass everything" reverse proxy implementation resembles the following. In this case, the rule specifies that if the request does not match any in the 1-22 rule set, then proxy the request to http://iNotes.forMyMailServer.Web.

Index	Action	Request template	Replacement file path
23	Proxy	/*	http://iNotes.forMyMailServer.Web/*

This setup is risky, however, because it allows direct access to any resource on the Domino server accessible via HTTP. The following table contains a more strict set of Domino resources to expose through the Edge server. In the rule sets below, only content within the /mail* subdirectories is served. The rule is defined so that if sites have multiple mail subdirectories (for instance, /mail[1-3]), they are accounted for. If you want to restrict access to only a subset of mail databases, move these to a special folder such as /pubmail/*. It would not be practical or wise to create a rule for each user’s mail database. The other rules allow access to supporting content needed to provide the iNotes Web Access experience to the end user.

Index	Action	Request template	Replacement file path
1	Proxy	/mail*	http:// xxx.xxx.xxx.xxx/mail*
2	Proxy	/iNotes/*	http:// xxx.xxx.xxx.xxx /iNotes/*
3	Proxy	/inotes5/*	http:// xxx.xxx.xxx.xxx /inotes5/*
4	Proxy	/icons/*	http:// xxx.xxx.xxx.xxx /icons/*
5	Proxy	/domjava/*	http:// xxx.xxx.xxx.xxx /domjava/*
6	Proxy	/names.nsf	http:// xxx.xxx.xxx.xxx /names.nsf

One important rule to consider is the /names.nsf rule used for authentication. This allows access to the Domino Directory from the Internet, but none of the content is available other than the default folder list. This is a result of how Domino builds URLs. When a user with session authentication logs into Domino, the default login screen sends a request to /names.nsf?Login. The Edge server matches this request and passes it to Domino. If, for example, a user tries to open the Groups view with /names.nsf/Groups?OpenView or

/names.nsf/85255ed5006cafef852556d4006ca21c?OpenView, then both the Domino and Edge server requests fail because they do not match the rule. The user receives an error 403 message stating access is forbidden.

As we discussed earlier, these request templates can also be controlled by IP address. If the Edge server is set up to support multiple IP addresses (host names), then each of the IP addresses can have separate rules populating the Server IP Address or Host Name column. If the value is blank, then the rule applies to all requests regardless of the IP address. One caveat: If a host name is used, then requests made with IP address (for example http://xxx.xxx.xxx.xxx/URL) fail because it does not match the rule. It also does not match partial hosts, so the host name of the proxy does not equal proxy.formymailserver.web.

The remaining sections of the Edge server Administration interface allow for addition system configuration. However, the screens covered in the preceding section are all you need to have a functional server.

ibmproxy.conf
There are other options important to the operation of the Edge server that cannot be entered through the Web Administration interface. These have to be set directly through ibmproxy.conf, so fire up your favorite text editor and get ready. The good thing is that examples of these are already defined in the configuration file, but are simply commented out.

The first is the LogVirtualHostName entry. This is important if you're running a virtual server on the Edge system; it modifies the log file so that each log entry contains the requested host. If this setting is not included, then there is no easy way to tell which IP an end user requested. Set this to beginning (other available options are merged and none):

```
LogVirtualHostName beginning
```

The SignificantUrlTerminator setting tells the Edge server that this value should be treated as part of the base URL. In other words, because most Domino URLs contain “?” they are treated as query URLs that represent dynamic content and as such are not eligible for caching locally on the Edge server. This directive tells the Edge server to start after this point of the URL to look for dynamic content. Because Domino uses ?OpenImageResource and ?OpenElement to represent what would be considered standard static GIF files, they can now be treated as such with regards to the Edge server:

```
SignificantUrlTerminator ?OpenImageResource
SignificantUrlTerminator ?OpenElement
SignificantUrlTerminator /?OpenImageResource
SignificantUrlTerminator /?OpenElement
```

The ReversePass directive intercepts the standard redirection 302 responses from Domino and rewrites them to a new location. This new location should be the valid external URL name so that when the end user requests the renew redirected page, it is still valid:

```
ReversePass http://xxx.xxx.xxx.xxx/* http://proxy.formymailserver.web/*
```

The SendRevProxyName directive is also very important as it passes the host name of the reverse proxy server to Domino. This ensures responses from Domino are sources to the Edge server and not the end client. This allows the Edge server to control the destination of the final content:

```
SendRevProxyName yes
```

The SSLOnly directive forces the Edge server to only deal with SSL traffic. This is required if your site does not allow for any other external protocol:

```
SSLOnly ON
```

The SSLCertificate directive is necessary for virtual server sites. When users request secure content from the Edge server, they are handed back an SSL certificate from that Edge server. This certificate must match what users are requesting; otherwise, they receive an error message. The Edge server is capable of handling multiple SSL certificates and will hand out these certificates based upon the requested IP address. When users request proxy.formymailserver.web, Edge sends the correct certificate. If they request proxy1.formymailserver.web, they receive a different certificate. The other key point of this is that when SSL is used for virtual servers on the Edge server, each virtual server requires its own IP address. There is a concept of named virtual servers where the server looks at the host name and determines the action appropriate to the IP address. This type of setup does not work correctly in the SSL world, but is perfectly valid for non-SSL usage:

```
SSLCertificate xxx.xxx.xxx.xxx proxy.formymailserver.web
```

There are additional directives available from efixes or patches to the Edge server application. One of the more interesting features that became available after efix 9 is the ability to rewrite HTML content on the fly:

```
----Junction URL Rewrite Plug-in ----

ServerInit c:\PROGRA-1\IBM\edge\cp\lib\plugins\mod_rewrite\mod_rw.dll:modrw_init

Transmogrifier
c:\PROGRA-1\IBM\edge\cp\lib\plugins\mod_rewrite\mod_rw.dll:modrw_open:modrw_write:modrw_close:modrw_er
ror

JunctionRewrite on
```

This allows virtualization of Domino servers behind a single proxy server URL. For instance, the URL http://proxy.formymailserver.web/server1 can point to one Domino server and http://proxy.formymailserver.web/server2 can point to another. The rewrite plug-in works directly with the Proxy rules to determine how the information should be modified. Keep in mind this can only update HTML content; a reference to /mail/jdoe.nsf can be changed to /server1/mail/jdoe.nsf. Example rule sets are shown in the following table:

Index	Action	Request template	Replacement file path
23	Proxy	/server1/*	http://iNotes1.forMyMailServer.Web/*
24	Proxy	/server2/*	http://iNotes2.forMyMailServer.Web/*

Single sign-on
As the industry has progressed to more available information on demand, an increasingly important topic is SSO (single sign-on). The nice thing is that because both the single server and multi-server implementations of SSO in Domino are handled via cookies, no additional changes need to occur on the Edge server to support SSO. The one key component is that the associated domain (for example, formymailserver.web) needs to remain the same across the internal and external systems.

Conclusion

Implementing a reverse proxy solution with Domino can be an interesting and (dare we say it?) fun exercise. Given careful planning and working with the information we provide in this article, implementation should be reasonably straightforward. Proxy technology allows for more secure access to messaging resources than has ever been possible before and will help make you the talk of the next staff meeting. So the next time you are on the road in Tokyo, Dallas, or London, you can access your iNotes Web Access email with the confidence that you have protected your corporate computing environment with a robust and secure reverse proxy solution.

ABOUT THE AUTHORS

David Byrd is a Consulting IT Architect with IBM Software Services for Lotus from Atlanta, GA. David is fluent in virtually all areas of Lotus products and technologies ranging from C/C++ API development, application architectures, security architectures, and messaging architectures with heavy focus in enterprise-level messaging and directories. He has worked with Lotus Notes and Domino for the past 10 years and holds numerous certifications from Lotus, IBM, Redhat, Microsoft, and Novell.

Timothy Speed is an infrastructure and security architect for IBM Software Services for Lotus (ISSL). Tim has been involved in Internet and messaging security for the last ten years. He also participated with the Domino infrastructure at the Nagano Olympics and assisted with the Lotus Notes systems for the Sydney Olympics. His certifications include MCSE©, VCA (VeriSign Certified Administrator), Lotus Domino CLP Principal Administrator, and Lotus Domino CLP Principal Developer. Tim has also co-authored three books: *The Internet Security Guidebook*, ISBN: 0122374711, February, 2001, *The Personal Internet Security Guidebook*, ISBN: 0126565619, October, 2001 and *Enterprise Directory and Security Implementation Guide: Designing and Implementing Directories in Your Organization*, ISBN:0121604527. You can reach Timothy at **Tim_Speed@us.ibm.com**.