



# EXCELERATOR 2.2



## ***Feature Pack 3 Release Notes***

## Legal Notices

Volera, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Volera, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Volera, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Volera, Inc. reserves the right to make changes to any and all parts of Volera software, at any time, without any obligation to notify any person or entity of such changes.

This product may require export authorization from the U.S. Department of Commerce prior to exporting from the U.S. or Canada.

Copyright © 1997-2002 Volera, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent Nos. 5,870,739; 5,873,079; 5,884,304; 6,330,605.. Patents pending.

Volera, Inc.  
2211 North First Street  
San Jose, CA 95131-2021  
U.S.A.s

[www.volera.com](http://www.volera.com)

Excelerator 2.2 Feature Pack 3 Release Notes  
September 2002

**Online Documentation:** To access the online documentation for this and other Volera products, and to get updates, see [www.volera.com](http://www.volera.com).

**Volera Trademarks**

Volera is a trademark of Volera, Inc. in the United States and other countries.

**Third-Party Trademarks**

All third-party trademarks are the property of their respective owners.



# Contents

	<b>Introducing Feature Pack 3</b>	<b>7</b>
<b>1</b>	<b>Issues and Caveats</b>	<b>9</b>
	URL Overrides . . . . .	9
	Multiple Services on Same IP Address and Port. . . . .	9
	Synchronization Problems with System Controller 1.2. . . . .	9
	Secure Filling Is Not Supported in Beta 1 . . . . .	9
<b>2</b>	<b>Domain-Based Acceleration</b>	<b>11</b>
	Understanding Domain-Based Acceleration . . . . .	11
	Graphical Overview. . . . .	12
	Authentication Profiles and Secure Exceleator Are Supported . . . . .	12
	URL Overrides Are Not Required. . . . .	13
	Using Authentication Profiles . . . . .	13
	Using Secure Exceleator . . . . .	13
	The Domain Name and Nodes . . . . .	13
	Logging . . . . .	14
	Using the CLI . . . . .	14
	Creating a Domain-Based Accelerator. . . . .	14
	CLI Summary. . . . .	15
<b>3</b>	<b>Changes Introduced in Feature Pack 2</b>	<b>21</b>
	Extended Logging Enhancements . . . . .	21
	Standard Header Field Logging . . . . .	21
	Pseudo Header Field Logging . . . . .	28
	Filter Service Bypass. . . . .	33
	Critical Information Regarding this Feature. . . . .	33
	Overview . . . . .	36
	Preparing to Install the Filter Bypass Software . . . . .	37
	Installing the Filter Bypass Software . . . . .	38
	Configuring the Main Perl Script on the IIS Server . . . . .	39
	Configuring the IIS Server To Run the Perl Scripts. . . . .	40
	Changing Each Exceleator's Default Blocked Content Page . . . . .	40
	Setting Up Administrative Access to the List of Bypass Users . . . . .	41
	Adding Authorized Filter Bypass Users. . . . .	41
	Starting the Filtering Bypass Service . . . . .	42
	Running in a Hierarchy with Filtering on the Parent . . . . .	42

	Starting the Filtering Bypass Service Automatically . . . . .	43
	Tips for Non-Default Installations . . . . .	43
	Notes About Product Security . . . . .	43
<b>4</b>	<b>Changes Introduced in Feature Pack 1</b>	<b>45</b>
	NTLM authentication multiple domain support . . . . .	45
	Telnet security . . . . .	46
	Adapter-Specific Settings . . . . .	46
<b>5</b>	<b>Installing Excelerator 2.2 Feature Pack 3</b>	<b>47</b>
	Update Clones Both Before and After Installing. . . . .	47
	Installing the Feature Pack . . . . .	47

# Introducing Feature Pack 3

Feature Pack 3 contains the features summarized in [Table 1](#).

Table 1

Feature	Summary
Service Pack 1	<p>Feature Pack. 3 is based on the Excelerator 2.2 Service Pack 1 release.</p> <p>For more information on Service Pack 1, see <a href="#">Service Pack 1 Release Notes</a>.</p>
Domain-Based Acceleration	<p>You can create a single acceleration service for all the Web servers in a single domain.</p> <p>For more information, see <a href="#">“Domain-Based Acceleration” on page 11</a>.</p>
Extended Logging Enhancements	<p>The information that can be collected in extended log files has been expanded.</p> <p>For more information, see <a href="#">“Extended Logging Enhancements” on page 21</a>.</p>
Filter Service Bypass	<p>Filtering functionality has been enhanced so that access can be granted temporarily to sites that are otherwise blocked. This is especially useful for school situations, wherein a teacher can grant temporary access for students to sites that are normally blocked.</p> <p>For more information, see <a href="#">“Filter Service Bypass” on page 33</a></p>
HTTPS Access Control	<p>Excelerator’s Access Control feature has been enhanced to apply to both HTTP and HTTPS requests. This ensures that HTTPS sites can be blocked even for users who have authenticated to the site.</p>
NTLM authentication multiple domain support	<p>NTLM authentication profiles now support multiple domains.</p>
Telnet security	<p>You can now disable and re-enable Telnet access to a cache device.</p>

---

Feature	Summary
Adapter-Specific Settings	You can now set vendor-specific network adapter settings from the command line. Settings are passed through to the driver loading instructions in system configuration files.

---

# 1

## Issues and Caveats

### URL Overrides

URL overrides seem to have problems rewriting content after installing Feature Pack 3 when the content on an accelerated origin server changes.

NOTE: URL overrides are automatic with domain-based acceleration.

### Multiple Services on Same IP Address and Port

After installing Feature Pack 3, you must be careful to not assign more than one service on the same IP address and port combination. The system will not warn you of the conflict.

### Synchronization Problems with System Controller 1.2

Feature Pack 3 will not currently synchronize with System Controller 1.2.

### Secure Filling Is Not Supported in Beta 1

If you are using Secure Excelsator with a domain-based acceleration service, be aware that secure filling between the domain acceleration service and the origin servers is not supported in the Beta 1 release of Feature Pack 3.



# 2

## Domain-Based Acceleration

In Feature Pack Three, Excelerator's appliance-based multihoming functionality has been extended to support Domain acceleration.

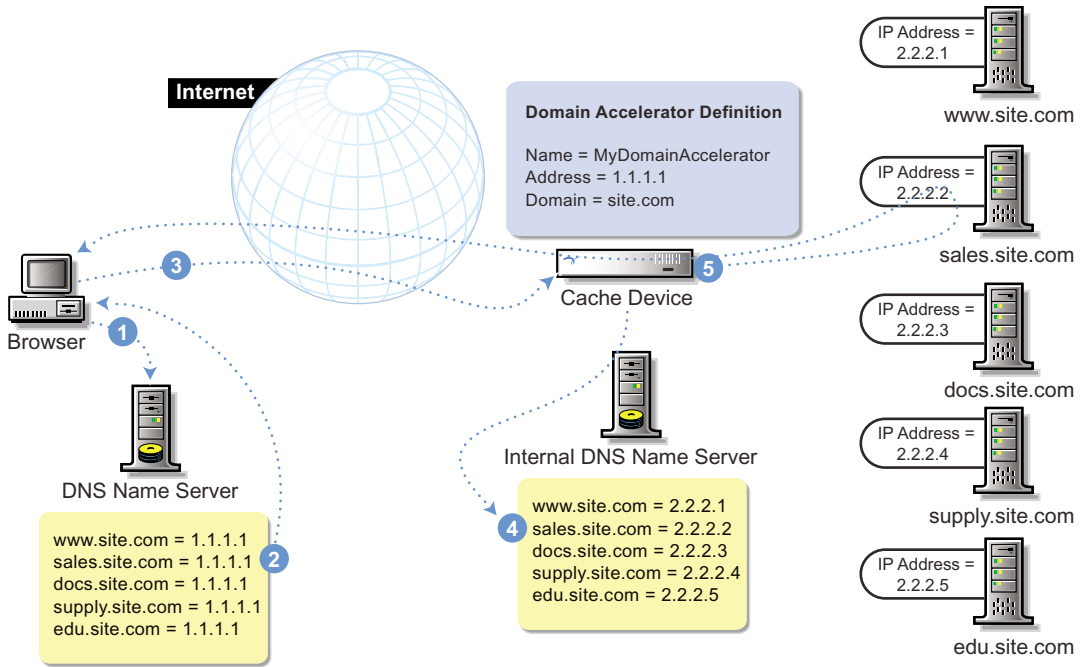
### Understanding Domain-Based Acceleration

In the past, a separate accelerator service had to be created for each Web server being accelerated. Now, domain-based multihoming lets you create a single domain-based accelerator to accelerate all the Web servers in a single domain.

# Graphical Overview

Figure 1 presents a basic overview of a single Exceleator cache device accelerating five Web servers in a single domain through a single domain-based acceleration service.

Figure 1



- 1 A browser requests an IP address for `sales.site.com`.
- 2 The DNS Name Server resolves `sales.site.com` to IP address `1.1.1.1` and returns the address to the browser.
- 3 The browser requests the objects from `sales.site.com` at IP address `1.1.1.1`.
- 4 The cache device receives the request, and using its internal DNS server obtains the IP address for the `sales.site.com` Web server.
- 5 The cache device obtains the objects from the Web server (if they aren't already in its cache) and returns them to the browser.

## Authentication Profiles and Secure Exceleator Are Supported

Domain-based multihoming also works with authentication profiles and Secure Exceleator implementations.

## URL Overrides Are Not Required

Domain-based accelerators automatically change cross-referenced URLs between Web servers within a domain. This eliminates the need to create URL overrides.

## Using Authentication Profiles

We recommend you create authentication profiles using the browser-based tool. Once the profiles have been created, you can assign them to domain-based accelerators using the CLI. See [Table 2 on page 15](#) for a list of CLI commands.

Once assigned, the profiles work as described for Web acceleration services.

## Using Secure Excelerator

Secure Excelerator interacts with domain-based acceleration services in basically the same way as with Web server-based acceleration services. For an overview of Web server acceleration functionality, see [Accelerating Web Servers](#) in the *Volera Excelerator 2.2 Administration Guide*.

In addition, there are two important differences:

- ◆ Domain-based acceleration services do not require the creation of URL overrides for links between servers within the domain being accelerated
- ◆ Domain-based acceleration services can only use one trusted root for encryption. Therefore, to use the secure fill feature with a domain acceleration service, you must ensure that every Web server in the domain is using the same trusted root as the acceleration service.

For more information on certificate maintenance, see [Managing Appliance Certificates](#) in the *Volera Excelerator 2.2 Administration Guide*.

## The Domain Name and Nodes

Domain-based accelerator services can handle up to five nodes before the domain name.

For example, an internal Web server with the domain name `svc.class1.edu.utah.orem.volera.com` has five nodes before the domain name `volera.com`.

## Logging

You must use the CLI to configure logging for domain-based acceleration services. However, once the log files are being created, you can use the browser-based tool to manage the downloading and pushing of log files. See [Table 2 on page 15](#) for a list of CLI commands.

NOTE: Log files for domain-based acceleration services are listed in the REVERSE log grouping in the browser-based tool along with other accelerator service logs.

## Using the CLI

[Table 2 on page 15](#) summarizes all the commands associated with domain-based acceleration.

The following sections outline some general tips for working with this feature.

## Creating a Domain-Based Accelerator

Complete the following steps:

- 1 Install Feature Pack 3 by completing the instructions in [Chapter 5, “Installing Excelerator 2.2 Feature Pack 3,”](#) on page 47.
- 2 At the system prompt, create a domain accelerator using the following command:  

```
add domainaccelerator = name
```

where *name* is the name you will use for managing the accelerator service.
- 3 Designate an IP address for handling incoming browser requests using the following command:  

```
set domainaccelerator name address = iii.iii.iii.iii
```

where *name* is the accelerator name you specified in [Step 2](#) and the *i*'s are a dot-delimited IP address.
- 4 Set one or more port numbers for the service using the following command:  

```
set domainaccelerator name port = port
```

where *name* is the accelerator name you specified in [Step 2](#) and *port* is the port number the service will listen on for incoming browser requests. If

you are using Secure Exceleator, you will need to specify both ports for both HTTP and HTTPS traffic (for example, port 80 and port 443).

IMPORTANT: You must ensure that no other services on the cache device are using the same IP address and port combinations.

- 5 Specify the internal domain name from which the accelerator service will fill incoming browser requests, by entering the following command:

```
set domainaccelerator name domainname = domain
```

where *name* is the accelerator name you specified in [Step 2](#) and *domain* is the name of the domain from which the service will fill browser requests.

- 6 Save the settings by entering

```
apply
```

## CLI Summary

The following table summarizes the commands used to set up and configure domain-based authentication. For detail on individual commands, refer to the CLI help.

Table 2

Feature	CLI Command
Basic Service	add domainaccelerator = <i>name</i>
	set domainaccelerator <i>name</i> enable = yes   no
	set domainaccelerator <i>name</i> domainname = <i>domain</i>
	set domainaccelerator <i>name</i> port = <i>port</i>
	set domainaccelerator <i>name</i> address = <i>address</i>
	set domainaccelerator <i>name</i> sendxforwardedforheader = yes   no
	set domainaccelerator <i>name</i> initialtcpreceivewindowsize = <i>size</i>
	set domainaccelerator <i>name</i> limitfillbandwidth = yes   no
	set domainaccelerator <i>name</i> cacheobjectswithnovalidatororexpirationdate = yes   no
	set domainaccelerator <i>name</i> browsernocacherequests = revalidate   ignore   refill

Feature	CLI Command
	set domainaccelerator <i>name</i> sslport = <i>port</i>
	set domainaccelerator <i>name</i> sslkeyid = <i>cert_name</i>
Authentication	set domainaccelerator <i>name</i> authentication enable = yes   no
	set domainaccelerator <i>name</i> authentication profile = <i>profile</i>
	set domainaccelerator <i>name</i> authentication sslmaxidletime = <i>time</i>
	set domainaccelerator <i>name</i> authentication cookiedomain = <i>domain</i>
	set domainaccelerator <i>name</i> authentication profilerule = and   or
	set domainaccelerator <i>name</i> authentication whenacfails = no   yes
	set domainaccelerator <i>name</i> authentication requiredonpost = no   yes
	set domainaccelerator <i>name</i> authentication requiredonput = no   yes
	set domainaccelerator <i>name</i> authentication requiredondelete = no   yes
	set domainaccelerator <i>name</i> authentication requiredontrace = no   yes
	set domainaccelerator <i>name</i> authentication requiredonoptions = no   yes
	set domainaccelerator <i>name</i> authentication requiredonconnect = no   yes
	set domainaccelerator <i>name</i> authentication requiredonhttp = no   yes
	set domainaccelerator <i>name</i> authentication requiredonother = no   yes
Referer Header Control	set domainaccelerator <i>name</i> referercontrol enable = yes   no
	set domainaccelerator <i>name</i> referercontrol url = <i>url</i>
Access Control	set domainaccelerator <i>name</i> accesscontrol enable = yes   no
	set domainaccelerator <i>name</i> accesscontrol policy = <i>policy</i>
Secure Exceleator	set domainaccelerator <i>name</i> securex enable = yes   no
	set domainaccelerator <i>name</i> securex secureaccesstowebserver = yes   no
	set domainaccelerator <i>name</i> securex trustedrootsfilelist = <i>list</i>

Feature	CLI Command
	set domainaccelerator <i>name</i> securex interactivehandshake = yes   no set domainaccelerator <i>name</i> securex acceptanyroot = yes   no set domainaccelerator <i>name</i> securex allowcacheatbrowser = yes   no
Custom Cache Header	set domainaccelerator <i>name</i> customcacheheaderenable = yes   no set domainaccelerator <i>name</i> customcacheheader = <i>header</i> set domainaccelerator <i>name</i> customcacheheaderonobjjexpiry = checkmodified   getnew
Logging - Common Format	set domainaccelerator <i>name</i> comlog = yes   no set domainaccelerator <i>name</i> comlogrolloption = bysize   bytime set domainaccelerator <i>name</i> comlogrollsize = <i>size</i> set domainaccelerator <i>name</i> comlogrollperiod = <i>minutes</i> set domainaccelerator <i>name</i> comlogrollday = 1stofmonth   monday   tuesday . . . set domainaccelerator <i>name</i> comlogrolltime = 12am   1 am   2 am . . . set domainaccelerator <i>name</i> comlogrolltimezone = local   gmt set domainaccelerator <i>name</i> comlogdeleteoption = nodelete   bytime   byfiles set domainaccelerator <i>name</i> comlogdeletemaxtime = <i>hours</i> set domainaccelerator <i>name</i> comlogdeletemaxfile = <i>files</i>
Logging - Extended Format	set domainaccelerator <i>name</i> extlog = yes   no set domainaccelerator <i>name</i> extlogrolloption = bysize   bytime set domainaccelerator <i>name</i> extlogrollperiod = <i>minutes</i> set domainaccelerator <i>name</i> extlogrollday = 1stofmonth   monday   tuesday . . . set domainaccelerator <i>name</i> extlogrolltime = 12am   1 am   2 am . . .

Feature	CLI Command
	set domainaccelerator <i>name</i> extlogrolltimezone = local   gmt
	set domainaccelerator <i>name</i> extlogrollsize = <i>size</i>
	set domainaccelerator <i>name</i> extlogdeleteoption = nodelete   bytime   byfiles
	set domainaccelerator <i>name</i> extlogdeletemaxtime = <i>hours</i>
	set domainaccelerator <i>name</i> extlogdeletemaxfile = <i>files</i>
	set domainaccelerator <i>name</i> extlogusername = yes   no
	set domainaccelerator <i>name</i> extlogservername = yes   no
	set domainaccelerator <i>name</i> extlogserverip = yes   no
	set domainaccelerator <i>name</i> extlogsitename = yes   no
	set domainaccelerator <i>name</i> extlogmethod = yes   no
	set domainaccelerator <i>name</i> extloguri = yes   no
	set domainaccelerator <i>name</i> extloguristem = yes   no
	set domainaccelerator <i>name</i> extloguriquery = yes   no
	set domainaccelerator <i>name</i> extloghttpversion = yes   no
	set domainaccelerator <i>name</i> extloghttpstatus = yes   no
	set domainaccelerator <i>name</i> extlogbytessent = yes   no
	set domainaccelerator <i>name</i> extlogbytesreceived = yes   no
	set domainaccelerator <i>name</i> extlogtimetaken = yes   no
	set domainaccelerator <i>name</i> extloguseragent = yes   no
	set domainaccelerator <i>name</i> extlogcookie = yes   no
	set domainaccelerator <i>name</i> extlogreferer = yes   no
	set domainaccelerator <i>name</i> extlogcachedstatus = yes   no
	set domainaccelerator <i>name</i> extlogfillproxyip = yes   no
	set domainaccelerator <i>name</i> extlogoriginip = yes   no

---

Feature	CLI Command
	set domainaccelerator <i>name</i> extlogxfwdfor = yes   no
	set domainaccelerator <i>name</i> extlogcontentrange = yes   no
	set domainaccelerator <i>name</i> extlogrange = yes   no
	set domainaccelerator <i>name</i> extlogifrange = yes   no
	set domainaccelerator <i>name</i> extlogcontentlength = yes   no
	set domainaccelerator <i>name</i> extlogetag = yes   no
	set domainaccelerator <i>name</i> extlogrequestpragma = yes   no
	set domainaccelerator <i>name</i> extlogreplypragma = yes   no
	set domainaccelerator <i>name</i> extlogcompleted = yes   no
	set domainaccelerator <i>name</i> extlogheadersize = yes   no
	set domainaccelerator <i>name</i> extlogcacheinfo = yes   no

---



# 3

## Changes Introduced in Feature Pack 2

The following enhancements introduced in Feature Pack 2 are also contained in Feature Pack 3.

### Extended Logging Enhancements

Feature Pack 2 included the following extended logging enhancements.

#### Standard Header Field Logging

The following standard header fields are now supported.

NOTE: *Sc* refers to server-to-client response headers, and *cs* refers to client to server request headers. Header content is included in double quotation marks, with null content indicated by a pair of marks.

#### **sc-(Content-Range)**

##### **Purpose**

This field records the byte ranges vended from an Exceleator cache device to a requesting client.

##### **Values:**

The results include the word *bytes*, are enclosed in double quotes ("), and are either one or more byte ranges separated by commas (,) or a null value ("") if this field is not applicable to the specific log entry (i.e. this is not a byte range request).

NOTE: A client may request one or more byte ranges in a single request and subsequently, a server may respond with one or more ranges.

In the case of a single range being vended, the log entry will contain an entry equivalent to the HTTP Content-Range header.

In the case of a multi-part response from the server, there are multiple Content-Range headers, which are found preceding each byte range. For multi-part range responses, the log entry will contain each of the ranges vended separated by commas, followed by a slash "/" and the total size of the object.

### **Enabling**

To enable this option, enter the following at the command line:

```
set accelerator name extlogcontentrange = yes  
apply
```

where *name* is the name of your web accelerator service.

### **Disabling**

To disable this option, enter the following at the command line:

```
set accelerator name extlogcontentrange = no  
apply
```

where *name* is the name of your web accelerator service.

### **Getting Status**

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogcontentrange
```

where *name* is the name of your Web Accelerator service.

### **Examples**

```
"bytes 100-200/2000","bytes 500-700,952-999/2000",""
```

## **cs(Range)**

### **Purpose**

This field records the byte ranges requested by a client to the Excelerator cache device.

## Values

The results include the word *bytes*, are enclosed in double quotes ("), and are either one or more byte ranges separated by commas (,) or a null value ("") if this field is not applicable to the specific log entry (i.e. this is not a byte range request).

## Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogrange = yes
apply
```

where *name* is the name of your Web Accelerator service.

## Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogrange = no
apply
```

where *name* is the name of your Web Accelerator service.

## Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogrange
```

where *name* is the name of your Web Accelerator service.

## Examples

```
"bytes 100-200", "bytes 500-700, 952-999", ""
```

## cs(If-Range)

### Purpose

This field records whether the client request was a conditional range request.

### Values

The result is enclosed in double quotation marks (") and is either the text of the range request or a null value ("") if this field is not applicable to the specific log entry (i.e. this is not a conditional byte range request).

NOTE: The value is typically either an Etag or a date.

### **Enabling**

To enable this option, enter the following at the command line:

```
set accelerator name extlogifrange = yes  
apply
```

where *name* is the name of your Web Accelerator service.

### **Disabling**

To disable this option, enter the following at the command line:

```
set accelerator name extlogifrange = no  
apply
```

where *name* is the name of your Web Accelerator service.

### **Status**

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogifrange
```

where *name* is the name of your Web Accelerator service.

### **Examples**

```
"Etag1234", "Tue, 17 Oct 2002", ""
```

## **sc(Content-Length)**

### **Purpose**

This field records the size (in bytes) of the entire object being vended from an Excelerator cache device to a requesting client.

### **Values**

The result is contained in double quotes (") and is either the content-length in bytes or null (") if this field is not applicable to the specific log entry (i.e. the total size of the object is unknown).

NOTE: A null value (") in the field might indicate that the object is not in cache and has to be piped from the origin server.

## Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogcontentlength = yes  
apply
```

where *name* is the name of your Web Accelerator service.

## Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogcontentlength = no  
apply
```

where *name* is the name of your Web Accelerator service.

## Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogcontentlength
```

where *name* is the name of your Web Accelerator service.

## Examples

```
"13245", "7776", ""
```

## sc(Etag)

### Purpose

This field records the Etag vended from an Exceleator cache device to a requesting client.

### Values

The result is enclosed in double quotes (") and is either the etag-text or null ("") if this field is not applicable to the specific log entry (i.e. no Etag provided).

NOTE: Etags may contain quotes, which will appear as "" to indicate that they are part of the field rather than the end of the field. For example, a log entry of ""123abc"" indicates an Etag value of "123abc". Etags may be used in conjunction with If-Range or other conditional requests.

## Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogetag = yes  
apply
```

where *name* is the name of your Web Accelerator service.

## Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogetag = no  
apply
```

where *name* is the name of your Web Accelerator service.

## Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogetag
```

where *name* is the name of your Web Accelerator service.

## Examples

```
""132mydoghasflees45"", "w/"123hiabc"", ""
```

## cs(Pragma)

### Purpose

This field records the pragma value associated with a client request to an Exceleator cache device.

### Values

The result is enclosed in double quotes (") and is either the pragma text value or null (""") if this field is not applicable to the specific log entry (i.e. a pragma header is not used).

NOTE: For HTTP 1.1 no-cache is the only valid value for pragma and although pragma: no-cache is intended for client to server requests, it has been implemented in practice in server to client responses as well.

## Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogrequestpragma = yes  
apply
```

where *name* is the name of your Web Accelerator service.

## Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogrequestpragma = no  
apply
```

where *name* is the name of your Web Accelerator service.

## Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogrequestpragma
```

where *name* is the name of your Web Accelerator service.

## Examples

```
"no-cache", ""
```

## sc(Pragma)

### Purpose

This field records the pragma value associated with a server response from an Excelerator cache device to a requesting client.

### Values:

The result is enclosed in double quotation marks (") and contains either the pragma text value or is blank if the field is not applicable to the specific log entry (i.e. a pragma header is not used)

NOTE: For HTTP 1.1 no-cache is the only valid value for pragma and although pragma: no-cache is intended for client to server requests, it has been implemented in practice in server to client responses as well.

## Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogreplypragma = yes  
apply
```

where *name* is the name of your Web Accelerator service.

## Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogreplypragma = no  
apply
```

where *name* is the name of your Web Accelerator service.

## Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogreplypragma
```

where *name* is the name of your Web Accelerator service.

## Examples

```
"no-cache", ""
```

## Pseudo Header Field Logging

The following pseudo header fields are also supported.

### sc-completed

#### Purpose

This field either records that a transaction was completed successfully or it indicates the reason for a failure to complete.

#### Values

The field contains one of the following values:

Success	Indicates either a normal connection termination or that the transaction was completed and the persistent connection remains open
Timeout	Indicates the connection timed out, perhaps due to network failure, and Excelerator discarded the connection
Reset	Indicates the connection was terminated by the client
Administrative	indicates that Excelerator terminated the connection for reasons unrelated to the connection, such as Excelerator shutting down
-	Not applicable to the specific log entry

### Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogcompleted = yes
apply
```

where *name* is the name of your Web Accelerator service.

### Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogcompleted = no
apply
```

where *name* is the name of your Web Accelerator service.

### Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogcompleted
```

where *name* is the name of your Web Accelerator service.

### Examples

Success, Reset, -

## sc-header-size

### Purpose

This field records the size (in bytes) of the HTTP header associated with a response to a client.

### Values

The field will contain either a byte-count of the record size or a dash (-) if this field is not applicable to the specific log entry.

HINT: To determine whether an entire object was downloaded, subtract this data from sc-bytes and compare the difference to the entity size.

### Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogheadersize = yes  
apply
```

where *name* is the name of your web accelerator service.

### Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogheadersize = no  
apply
```

where *name* is the name of your web accelerator service.

### Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogheadersize
```

where *name* is replaced with the name of your Web Accelerator service.

### Examples

678, 679, -

## x-cache-info

### Purpose

This field records brief status statements for objects vended from cache or brief reasons why a requested object was not cached.

### Values

The following statements or reasons are enclosed in double quotation marks (""). Additional information is enclosed in parentheses ().

pass thru

error

redirect

in cache

fresh

revalidate each use

stale

not modified

updated

not in cache

filled

if revalidate each use (The reasons it was revalidate each use)

not cacheable (The reason is was not cacheable)

NOTE: This information could be very useful in troubleshooting.

### Enabling

To enable this option, enter the following at the command line:

```
set accelerator name extlogcacheinfo = yes  
apply
```

where *name* is the name of your Web Accelerator service.

## Disabling

To disable this option, enter the following at the command line:

```
set accelerator name extlogcacheinfo = no  
apply
```

where *name* is the name of your Web Accelerator service.

## Getting Status

To get logging status for this option, enter the following at the command line:

```
get accelerator name extlogcacheinfo
```

where *name* is the name of your Web Accelerator service.

## Examples

"In Cache, Browser Requested Revalidate, Not Modified", "Not In Cache, Not Cacheable (No Validator Or Expiration Time)", "In Cache, Fresh"

# Filter Service Bypass

Filtering functionality was enhanced in Feature Pack 2 so that an N2H2 filtering service or an override list entry can be temporarily bypassed, allowing objects to be cached that would otherwise be blocked.

NOTE: The filter service bypass functionality does not apply to installed X-Stop filtering services.

This feature can be useful for school situations wherein a teacher can either cache content or permit content to be cached that would normally be blocked. For example, sites dealing with breast cancer research might normally be blocked, in which case a teacher could cause the sites to be cached and make the information available to students writing research papers.

## Critical Information Regarding this Feature

Before deploying this feature, it is critical that you understand exactly how it works as outlined in the following sections.

### Understanding How the Feature Works

Consider the following points:

- ♦ **Blocking Prevents Objects from Being Cached:** Excelsior filtering services and override lists block specified Web content from coming through the proxy service. Content that can't get through the proxy service can't be cached.

Conversely, if content is not blocked and if it is cachable, it is cached automatically as part of the retrieval process.

- ♦ **Filtering Services Prevent Access Only Where Installed:** An installed filtering service or an override list will prevent general access to cached content on the device where they are installed or active.

However, they do not prevent general access to cached content on other devices. For example, a filtering service on a hierarchical parent will not prevent access to cached content on a hierarchical child that doesn't have a filtering service installed.

- ♦ **Unblocked Cached Content Is Openly Available:** Once content is cached, anyone accessing the Web through the cache device has access to the cached content if a filtering service or override list on the box doesn't block the access.

- ◆ **Hierarchical Caching Broadens the Distribution of Cached Objects:** When a cachable object is requested and retrieved through a hierarchy, any hierarchical parents or peers involved in retrieving the objects will also have cached copies of the objects.

Therefore, anyone accessing the Web through these cache devices will also have access to the objects unless they are blocked by a filtering service or override list on the device.

- ◆ **Unblocked Access Is Unrestricted Access:** Once filtering has been bypassed for a workstation, anyone using the workstation has unrestricted Web access for the period of time specified when access is granted.

It is imperative, therefore, that unblocked access is carefully monitored and controlled.

### What the Feature Doesn't Provide

- ◆ **Access Control to the Web:** After a workstation is authorized to bypass filtering, Web access is unrestricted.
- ◆ **Object Caching Controls:** Once filtering is bypassed, all cachable objects accessed through the cache device will be cached.
- ◆ **Access to Cached Objects:** Once objects are cached, they become generally available unless one of the following happens:
  - ◆ A filtering service or override list *on the box* blocks access
  - ◆ The objects expire
  - ◆ The objects are purged from cache

### Hierarchical Considerations

Organizations often set up hierarchies wherein all content requests are funneled through one or two parent cache devices. Such organizations usually install filtering services only on the parent devices since these can then act as gatekeepers for all content entering the hierarchy.

Obviously, when filtering is bypassed and content that would normally be blocked is instead brought into the hierarchy, the scenario changes substantially. Requests made directly to the parent devices with filtering installed will still be blocked. However, requests to devices that don't have filtering installed will be filled directly from cache.

## Managing the Impacts of Using this Feature

Organizations generally deploy filtering services and override lists to ensure that only appropriate content is cached and made generally available. For this reason, any decisions to bypass these safeguards must be carefully considered.

We recommend you consider the following usage suggestions:

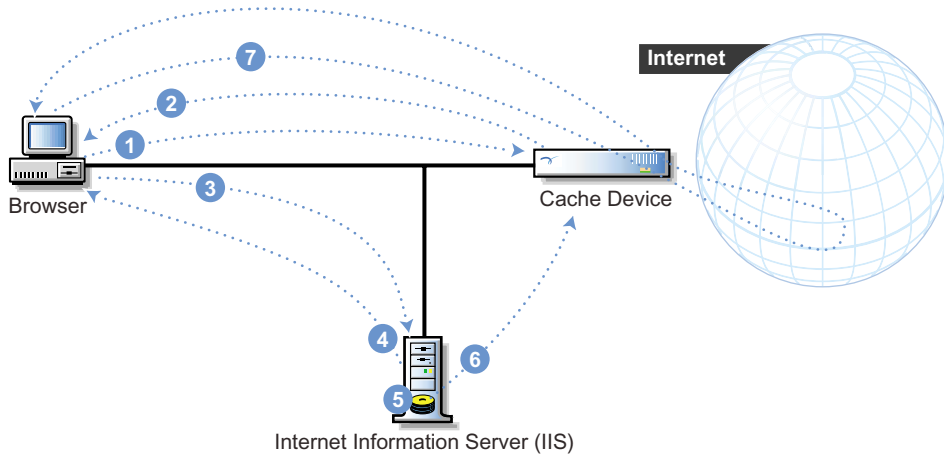
1. Use the feature only as a means for caching content needed for specific purposes. For example, a teacher might bypass filtering to access sites on breast cancer research. If site objects are cachable, the information will be cached and become generally available to everyone accessing the Web through the cache device, as long as a filtering service or override list on the device doesn't block access.
2. In caching hierarchies, consider that content will be cached on all parents and peers involved in retrieving objects. Evaluate whether the content is appropriate to other groups who will access the cache devices that don't have filtering services or override lists. If the content is not appropriate, perform a purge on the devices that the other groups will access.
3. When the cached content is no longer needed, use a selective purge to remove it from cache.

**IMPORTANT:** If undesirable content is accessed and cached for whatever reason, you will need to use the selective purge to remove it from the cache device and any hierarchical parents or peers involved in retrieving the content.

# Overview

Filter bypass functionality is summarized in **Figure 2**.

Figure 2



- 1 A user requests access to a Web page through a service on a cache device with N2H2 installed or an override list defined.
- 2 N2H2 or the override list indicates that the Web page is blocked. The device then sends the browser an Access Denied page containing a form for requesting permission to bypass the block.
- 3 The user provides a user id, password, and bypass duration, and the browser sends this to the Internet Information Server (IIS).
- 4 If the information is verified, the IIS notifies the browser that the request has been granted and the process continues with Step 5.

If the information is not verified, the process repeats from Step 2 until an unblocked URL is requested.

- 5 The IIS adds a line to its bypass file that contains all IP addresses currently authorized to bypass filtering and override blocks.
- 6 Within 60 seconds, IIS transmits an updated bypass list to the device through FTP.
- 7 After waiting up to 60 additional seconds, the user has unrestricted Web access. Any content accessed is cached.

After the specified bypass duration expires, the user no longer has unrestricted access. However, on any devices that are members of hierarchies and that don't have filtering services or override lists, content that has been cached is generally available until it either expires or is purged from cache.

## Preparing to Install the Filter Bypass Software

To deploy the filter bypass solution, you need the following:

- ◆ At least an intermediate understanding of the following:
  - ◆ Excelerator proxy services
  - ◆ N2H2 filtering on Excelerator
  - ◆ General networking
  - ◆ Microsoft Internet Information Server Web services
- ◆ Feature Pack 2 installed on one or more Excelerator 2.2 cache devices
- ◆ N2H2 filtering software or override lists configured and active on the cache devices
- ◆ A Windows 2000 computer with at least a Pentium III 500 Mhz processor and a minimum of 256 MB RAM installed
- ◆ A static (non-DHCP), publicly accessible IP address assigned to the Windows 2000 computer.
- ◆ Microsoft Internet Information Server (IIS) version 5.0 or later installed and running as a Web server on the Windows 2000 machine
- ◆ Active State's ActivePerl 5.6.x or later installed and running on the Windows 2000 IIS computer

Complete the following steps:

- 1 Install Feature Pack 2 on each affected Excelerator 2.2 cache device by completing the instructions in [“Installing Excelerator 2.2 Feature Pack 3” on page 47](#).
- 2 Install and configure an N2H2 filtering service or create an override list on each Excelerator 2.2 cache device.

For help installing N2H2 filtering services, refer to the N2H2 product documentation. For more information on creating filter override lists, see [Creating an Override List](#) in the *Volera Excelerator 2.2 Administration Guide*.

- 3 Enable the Mini-FTP server on each Excelerator cache device.

For more information, see [FTP Services](#) in the *Volera Excelerator 2.2 Administration Guide*.

- 4 Install the Microsoft Internet Information Server (IIS) on the Windows 2000 computer.

This software is available on the Windows 2000 CD if one came with the computer or on [Microsoft's Web site \(http://www.microsoft.com/\)](http://www.microsoft.com/).

- 5 Download and complete a default installation of ActiveState's ActivePerl 5.6.x or later from [ActiveState's Web site \(http://www.activestate.com\)](http://www.activestate.com).

NOTE: The download is free and should be installed using the default options for use with Microsoft IIS as outlined in the accompanying documentation.

If the default options are not used, for example if you install in another location, be sure to follow the tips presented in "[Tips for Non-Default Installations](#)" on page 43.

- 6 Ensure that the Perl executable appears in the path on the Windows 2000 computer by running CMD.EXE and entering PATH at the command prompt.

## Installing the Filter Bypass Software

Complete the following steps:

- 1 Insert the Feature Pack 2 CD in the Windows 2000 computer's CD drive.
- 2 Browse to the Support directory on the CD > run N2H2FBYP.EXE > unzip the files to the default location.
- 3 Run the command processor (CMD) from the Start menu, and using the CD command, change to Documents and Settings\*username*\Local Settings\Temp, where *username* is your Windows 2000 user name.

NOTE: Local Settings is a hidden directory and won't appear in a directory listing (DIR) at the command line. However, the directory is accessible and the CD command should work.

- 4 In the command processor window, enter INSTALL.BAT.

This batch file copies the following files to the directories indicated:

- ♦ The root of the IIS Web server (default is C:\inetpub\wwwroot)  
Adduser.html
- ♦ The IIS Web server's cgi-bin directory (default is C:\inetpub\wwwroot\cgi-bin)  
adduser.pl  
login.pl  
login.htx

logout.pl  
logout.html

- ◆ The filter work directory (default is C:\filter)
  - resetAdminPass.pl
  - configex.pl
  - sysAdminPassword.dat

## Configuring the Main Perl Script on the IIS Server

After installing the filter bypass software on the IIS server, you must modify its Perl scripts to communicate with your Exceleator cache devices.

Complete the following steps:

- 1 Using an ASCII text editor, open the configex.pl file in the C:\filter directory.

NOTE: This is the main Perl script that communicates with the Exceleator cache device.

- 2 Locate the following line in the file:

```
$uploadServers = "<Mini-FTP_IP_Address>:config:<Pwd>";
```

This line specifies the Exceleator Mini-FTP servers to which the IIS server will send bypass list files.

IMPORTANT: In the steps that follow, modify only the text between the double quotation marks ("...").

- 3 Replace *<Mini-FTP\_IP\_Address>* with the IP address of one of the Exceleators' mini-FTP servers to which you want the bypass lists sent.
- 4 Do not change the word *config* in the string. This references the Exceleator config user - the user with rights to copy files using FTP.
- 5 Replace the word *<Pwd>* with the password of the config user on the Exceleator.
- 6 If you want the bypass lists sent to multiple Exceleators, create new entries (*<Mini-FTP\_IP\_Address>:config:<Pwd>*) within the double quotation marks for each Exceleator and separate each entry using commas (,).

For example, "1.1.1.1:config:pwd1,2.2.2.2:config:pwd2, ... ".

- 7 If you have not installed all components to their default locations, complete the modifications outlined in “[Tips for Non-Default Installations](#)” on page 43.

## Configuring the IIS Server To Run the Perl Scripts

Complete the following steps to enable the IIS server to run Perl scripts located in the cgi-bin directory:

- 1 On the IIS server, click Start > Settings > Control Panel > Administrative Tools.
- 2 Run the Internet Services Manager.
- 3 Browse in the tree to the Default Web Site > cgi-bin folder.
- 4 Right-click on cgi-bin > select Properties.
- 5 Under Execute Permissions, select Scripts and Executables.

## Changing Each Excelerator’s Default Blocked Content Page

You must change the default HTML page that Excelerator serves when access to blocked content is attempted so that the completed form can be sent to the IIS. You must then FTP this page to each Excelerator.

Complete the following steps:

- 1 Browse to C:\Documents and Settings\*username*\Local Settings\Temp\misc where *username* is the Windows 2000 user name you used when installing the filtering bypass software, and make a copy of the picsblok.htm file.
- 2 Using an ASCII text editor, open the picsblok.htm file.
- 3 Locate the string <IP\_OF\_WEB\_SERVER> in the file and replace only this string with the IP address of the IIS server.

IMPORTANT: If you choose to modify other portions of the picsblok.htm file, *do not* modify the following lines:

```
<input type="hidden" name="REFERER" value="#REFERER#">
<input type="hidden" name="FORWARDED_FOR_IP" value="<X-
FORWARDED-FOR">
```

These two lines are critical elements of the form that cannot be changed or deleted.

- 4 Using an FTP application, copy the modified picsblok.htm file to the etc/proxy/data directory on the Exceleator cache device where filtering is installed.

## Setting Up Administrative Access to the List of Bypass Users

Access to the list of filter bypass users and their passwords is controlled by a single administrative password. To set or change this password, complete the following steps:

- 1 On the IIS server, run the command processor (CMD) from the Start menu.
- 2 Change to the C:\filter directory, or to the directory where the setadminpass.pl script file is located.
- 3 Enter the following command:

```
perl setadminpass.pl xxxxx
```

where xxxxx is the administrative password you want to use when modifying the filter bypass user database.

## Adding Authorized Filter Bypass Users

**IMPORTANT:** This section lets you designate the people who will be able to bypass filtering. In most situations, this will be a supervisor (or teacher in a school environment) who has the responsibility to see that only appropriate content is accessed and cached.

To add authorized users to the filter bypass database, complete the following steps:

- 1 In a browser, access the Add User page on the IIS by entering the following URL:  

```
http://iii.iii.iii.iii/AddUser.html
```

where *i* is the IP address of the IIS Web server.
- 2 In the SysAdmin Password field, type the administrative password created in [“Setting Up Administrative Access to the List of Bypass Users” on page 41](#).
- 3 In the UserName field, type the user id for one of the filter bypass users.
- 4 In the Password field, type the password this user will use to bypass filtering.

- 5 In the Verify Password field, re-type the password for this user.
- 6 Click Add User.
- 7 To create additional users, repeat the process starting with **Step 2**.

## Starting the Filtering Bypass Service

NOTE: All maintenance of bypass lists, expiration checking, etc. occurs on the IIS server. It is therefore critical that the server communicate regularly with the Excelerator cache devices. Otherwise, the devices won't receive timely bypass list updates, and the expiration of IP addresses will also not be communicated to the cache devices.

To start the IIS server sending bypass lists to your Excelerators, you need to run a Perl script on the server.

Complete the following Steps:

- 1 On the IIS server, run the command processor (CMD) from the Start menu.
- 2 Change to the C:\filter directory, or the directory containing the configex.pl Perl script.
- 3 Enter the following command:

```
perl configex.pl
```

By default, this script runs continually, processing new and expired bypass authorizations every 60 seconds. It also provides a status update every 60 seconds indicating that the script is still running.

## Running in a Hierarchy with Filtering on the Parent

If your users are accessing the Web through a proxy that is a member of a hierarchy in which filtering is installed on the parent cache device, you must do the following on the child device:

- ◆ Check the X-Forwarded-For option in the proxy service definition
- ◆ Check the Must Forward Through Hierarchy option in the hierarchy definition

IMPORTANT: Be aware that any objects cached on a device that fills through a hierarchy will also be cached on all other devices involved in retrieving the content. For a review of the implications of caching through a hierarchy, see ["Managing the Impacts of Using this Feature" on page 35](#).

## Starting the Filtering Bypass Service Automatically

IMPORTANT: You should start the bypass service from the command line at first and watch for any error messages or other issues requiring troubleshooting.

If you want to have the script start automatically each time the server reboots, you can use `srvany.exe` and `instsrv.exe`, available with the W2K Resource Kit, to convert `configex.pl` to an NT service. More information is available on the Web ([http://www.perlguy.com/articles/nt\\_service.html](http://www.perlguy.com/articles/nt_service.html)).

Alternatively, you can also use a third party CRON utility to run the script every 45 to 60 seconds.

## Tips for Non-Default Installations

If you have installed any components (ActivePerl, the bypass software, Internet Information Server, etc.) to non-default locations, you must check the `.html`, `.htx`, and `.pl` files in the following locations and modify any affected paths.

- ◆ `non_default_directory_path/filter`
- ◆ `non_default_directory_path/inetpub\wwwroot`
- ◆ `non_default_directory_path/inetpub\wwwroot\cgi-bin`

where `non_default_directory_path` represents the directory where you've installed the bypass software.

## Notes About Product Security

The filtering bypass software manages a simple database of userids and passwords for the users authorized to grant temporary unrestricted Web access, such as teachers who need to cache specific content in an educational environment.

Passwords in this database are stored in their hashed format, and the database is not stored in a path that is accessible to the HTTP server. A would-be hacker would have to compromise Windows 2000 security to crack the passwords in the database.

Creating the userids and passwords in the database is handled through the HTML input password mechanism. Anyone sniffing the wire would be able to see the password in plain text.

Access to the database requires an administrative password. For more information on setting this password, see [“Setting Up Administrative Access to the List of Bypass Users” on page 41](#). Since this password is created on the box, it cannot be sniffed on the wire. One would have to look over the administrator’s shoulder to discover this password.

The IIS server sends the bypass list to the Excelerator using Excelerator’s mini-FTP functionality. All FTP security liabilities apply to this transaction.

# 4

## Changes Introduced in Feature Pack 1

The following enhancements introduced in Feature Pack 1 are also included in Feature Pack 3.

### NTLM authentication multiple domain support

NTLM authentication profiles now support multiple domains.

The username and groupname strings used by Excelerator always include the domain name followed by a back slash (\). Therefore, log file entries will contain the domain combined with the other names.

Access control rules that refer to NTLM authentication profiles must now contain the domain name followed by the username or groupname.

The procedure for creating profiles has not changed. Multiple domain support is handled by the trust relationships between domains.

The installation instructions contained in the Excelerator 2.2 Administration Guide have not changed.

However, you will need to do the following

- 1 On each Domain Controller used by an existing NTLM authentication profile, install the new VOLAUTH.EXE file contained on the Feature Pack CD in the SUPPORT\_FILES directory.
- 2 Use the new VOLAUTH.EXE file when configuring additional Domain Controllers for NTLM authentication profiles.
- 3 Establish trust relationships between the Domain referenced in a profile and any other domains being used for authentication.

- 4 If you have previously created access controls that refer to NTLM-based authentication profiles, edit the controls and insert a domain name and back slash (\) before any usernames or groupnames included in the controls.
- 5 Include the domain name and back slash (\) with all usernames and groupnames included in new access controls you create.

## Telnet security

You can now disable and re-enable Telnet access to a cache device.

The action is immediately effective and any active Telnet connections are immediately terminated.

To disable Telnet access to a cache device, enter the following commands at the System prompt:

```
set listener telnet enable=no  
apply
```

To re-enable Telnet access, enter the following commands:

```
set listener telnet enable=yes  
apply
```

## Adapter-Specific Settings

You can now set vendor-specific network adapter settings from the command line. Settings are passed through to the driver loading instructions in system configuration files.

Enter the following at the System prompt:

```
set ethx loadlineparameters=parameter  
apply
```

where *x* is the adapter number and *parameter* is the vendor-specific adapter setting.

**IMPORTANT:** The system does not check for parameter setting errors. You must ensure that the exact syntax is entered.

To clear a setting, use the set command shown above without including a parameter.

# 5

## Installing Excelerator 2.2 Feature Pack 3

**IMPORTANT:** This product can only be installed on a cache device running Excelerator 2.2.56 or later.

Complete the instructions in the following sections to install Excelerator 2.2 Feature Pack 3.

### Update Clones Both Before and After Installing

We strongly recommend that you update the clone image both before and after installing the feature pack. You should wait to update the clone image after installing the feature pack until you are sure the appliance is behaving as expected.

To update the clone image, click **System > Actions > Update Clones** in the browser-based management tool.

After installing the feature pack, if you are unhappy with results for any reason, you can revert back to the previous system image.

To restore a clone image, click **System > Actions > Restore from Clones** in the browser-based management tool.

After you have verified the upgrade is performing as expected, you should then update the clone image to the feature pack version.

### Installing the Feature Pack

- 1 Insert the Feature Pack CD in a workstation and browse to the OTWUG directory.

- 2 Note the name of the .TXT file in the directory.
- 3 Insert the Feature Pack CD in the cache device.
- 4 On your workstation, start the device's browser-based management tool.
- 5 Click System > Upgrade.
- 6 Check Enable Download.
- 7 In the Install from URL field, type:  

```
file:///DOC_CD:OTWUG/filename.txt
```

where filename is the name of the text file noted in **Step 2**.
- 8 Click the Download Time drop-down list > select the time you want the install package copied from the CD to the cache device > check Enable Install.
- 9 Click the Install Time drop-down list > select the time you want the feature pack to be installed.
- 10 Click Apply.  

The process always restarts the device, temporarily disconnecting the browser-based management tool.
- 11 When the process completes, remove the Feature Pack CD and restart the device.