IBM Security Access Manager for Web
Version 7.0

IBM Security Web Gateway Appliance Administration Guide

SC22-5432-00

IBM Security Access Manager for Web
Version 7.0

IBM Security Web Gateway Appliance Administration Guide

SC22-5432-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 321.

Edition notice

Note: This edition applies to version 7, release 0, modification 0 of IBM Security Access Manager (product number 5724-C87) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

About this publication

Welcome to the *IBM Security Web Gateway Appliance Administration Guide*.

IBM Security Access Manager for Web, formerly called IBM Tivoli Access Manager for e-business, is a user authentication, authorization, and web single sign-on solution for enforcing security policies over a wide range of web and application resources.

IBM Security Access Manager for Web WebSEAL is the resource manager for Web-based resources in a Security Access Manager secure domain. WebSEAL is a high performance, multi-threaded Web server that applies fine-grained security policy to the protected Web object space. WebSEAL can provide single sign-on solutions and incorporate back-end Web application server resources into its security policy.

The IBM Security Web Gateway Appliance uses the strength of WebSEAL to provide access control and protection from Web-based threats.

Intended audience

This guide is for system administrators responsible for configuring and maintaining a Security Access Manager WebSEAL environment.

Readers should be familiar with the following:

- PC and UNIX or Linux operating systems
- Database architecture and concepts
- Security management
- Internet protocols, including HTTP, TCP/IP, File Transfer Protocol (FTP), and Telnet
- Lightweight Directory Access Protocol (LDAP) and directory services
- A supported user registry
- WebSphere® Application Server administration
- Authentication and authorization

If you are enabling Secure Sockets Layer (SSL) communication, you also should be familiar with SSL protocol, key exchange (public and private), digital signatures, cryptographic algorithms, and certificate authorities.

Access to publications and terminology

This section provides:

- A list of publications in the “IBM Security Access Manager for Web library” on page x.
- Links to “Online publications” on page xi.
- A link to the “IBM Terminology website” on page xii.

IBM Security Access Manager for Web library

The following documents are in the IBM Security Access Manager for Web library:

- *IBM Security Access Manager for Web Quick Start Guide*, GI11-9333-01

  Provides steps that summarize major installation and configuration tasks.
- *IBM Security Web Gateway Appliance Quick Start Guide* – Hardware Offering

  Guides users through the process of connecting and completing the initial configuration of the WebSEAL Hardware Appliance, SC22-5434-00
- *IBM Security Web Gateway Appliance Quick Start Guide* – Virtual Offering

  Guides users through the process of connecting and completing the initial configuration of the WebSEAL Virtual Appliance.
- *IBM Security Access Manager for Web Installation Guide*, GC23-6502-02

  Explains how to install and configure Security Access Manager.
- *IBM Security Access Manager for Web Upgrade Guide*, SC23-6503-02

  Provides information for users to upgrade from version 6.0, or 6.1.x to version 7.0.
- *IBM Security Access Manager for Web Administration Guide*, SC23-6504-02

  Describes the concepts and procedures for using Security Access Manager. Provides instructions for performing tasks from the Web Portal Manager interface and by using the **pdadmin** utility.
- *IBM Security Access Manager for Web WebSEAL Administration Guide*, SC23-6505-02

  Provides background material, administrative procedures, and reference information for using WebSEAL to manage the resources of your secure Web domain.
- *IBM Security Access Manager for Web Plug-in for Web Servers Administration Guide*, SC23-6507-02

  Provides procedures and reference information for securing your Web domain by using a Web server plug-in.
- *IBM Security Access Manager for Web Shared Session Management Administration Guide*, SC23-6509-02

  Provides administrative considerations and operational instructions for the session management server.
- *IBM Security Access Manager for Web Shared Session Management Deployment Guide*, SC22-5431-00

  Provides deployment considerations for the session management server.
- *IBM Security Web Gateway Appliance Administration Guide*, SC22-5432-00

  Provides administrative procedures and technical reference information for the WebSEAL Appliance.
- *IBM Security Web Gateway Appliance Configuration Guide for Web Reverse Proxy*, SC22-5433-00

  Provides configuration procedures and technical reference information for the WebSEAL Appliance.
- *IBM Security Web Gateway Appliance Web Reverse Proxy Stanza Reference*, SC27-4442-00

  Provides a complete stanza reference for the IBM® Security Web Gateway Appliance Web Reverse Proxy.
- *IBM Security Access Manager for Web WebSEAL Configuration Stanza Reference*, SC27-4443-00

  Provides a complete stanza reference for WebSEAL.

- *IBM Global Security Kit: CapiCmd Users Guide*, SC22-5459-00

  Provides instructions on creating key databases, public-private key pairs, and certificate requests.
- *IBM Security Access Manager for Web Auditing Guide*, SC23-6511-02

  Provides information about configuring and managing audit events by using the native Security Access Manager approach and the Common Auditing and Reporting Service. You can also find information about installing and configuring the Common Auditing and Reporting Service. Use this service for generating and viewing operational reports.
- *IBM Security Access Manager for Web Command Reference*, SC23-6512-02

  Provides reference information about the commands, utilities, and scripts that are provided with Security Access Manager.
- *IBM Security Access Manager for Web Administration C API Developer Reference*, SC23-6513-02

  Provides reference information about using the C language implementation of the administration API to enable an application to perform Security Access Manager administration tasks.
- *IBM Security Access Manager for Web Administration Java Classes Developer Reference*, SC23-6514-02

  Provides reference information about using the Java™ language implementation of the administration API to enable an application to perform Security Access Manager administration tasks.
- *IBM Security Access Manager for Web Authorization C API Developer Reference*, SC23-6515-02

  Provides reference information about using the C language implementation of the authorization API to enable an application to use Security Access Manager security.
- *IBM Security Access Manager for Web Authorization Java Classes Developer Reference*, SC23-6516-02

  Provides reference information about using the Java language implementation of the authorization API to enable an application to use Security Access Manager security.
- *IBM Security Access Manager for Web Web Security Developer Reference*, SC23-6517-02

  Provides programming and reference information for developing authentication modules.
- *IBM Security Access Manager for Web Error Message Reference*, GI11-8157-02

  Provides explanations and corrective actions for the messages and return code.
- *IBM Security Access Manager for Web Troubleshooting Guide*, GC27-2717-01

  Provides problem determination information.
- *IBM Security Access Manager for Web Performance Tuning Guide*, SC23-6518-02

  Provides performance tuning information for an environment that consists of Security Access Manager with the IBM Tivoli Directory Server as the user registry.

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Access Manager for Web Information Center
The http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.isam.doc_70/welcome.html site displays the information center welcome page for this product.

IBM Publications Center
The http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss site offers customized search functions to help you find all the IBM publications that you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

Related publications

This section lists the IBM products that are related to and included with the Security Access Manager solution.

Note: The following middleware products are not packaged with IBM Security Web Gateway Appliance.

IBM Global Security Kit

Security Access Manager provides data encryption by using Global Security Kit (GSKit) version 8.0.x. GSKit is included on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your particular platform.

GSKit version 8 includes the command-line tool for key management, GSKCapiCmd (**gsk8capicmd_64**).

GSKit version 8 no longer includes the key management utility, iKeyman (**gskikm.jar**). iKeyman is packaged with IBM Java version 6 or later and is now a pure Java application with no dependency on the native GSKit runtime. Do not move or remove the bundled *java*/jre/lib/gskikm.jar library.

The *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 and 7, iKeyman User's Guide for version 8.0* is available on the Security Access Manager Information Center. You can also find this document directly at:

http://download.boulder.ibm.com/ibmdl/pub/software/dw/jdk/security/60/iKeyman.8.User.Guide.pdf

Note:

GSKit version 8 includes important changes made to the implementation of Transport Layer Security required to remediate security issues.

The GSKit version 8 changes comply with the Internet Engineering Task Force (IETF) Request for Comments (RFC) requirements. However, it is not compatible with earlier versions of GSKit. Any component that communicates with Security Access Manager that uses GSKit must be upgraded to use GSKit version 7.0.4.42, or 8.0.14.26 or later. Otherwise, communication problems might occur.

IBM Tivoli Directory Server

IBM Tivoli Directory Server version 6.3 FP17 (6.3.0.17-ISS-ITDS-FP0017) is included on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your particular platform.

You can find more information about Tivoli Directory Server at:

http://www.ibm.com/software/tivoli/products/directory-server/

IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator version 7.1.1 is included on the *IBM Tivoli Directory Integrator Identity Edition V 7.1.1 for Multiplatform* product image or DVD for your particular platform.

You can find more information about IBM Tivoli Directory Integrator at:

http://www.ibm.com/software/tivoli/products/directory-integrator/

IBM DB2 Universal Database™

IBM DB2 Universal Database Enterprise Server Edition, version 9.7 FP4 is provided on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your particular platform. You can install DB2® with the Tivoli Directory Server software, or as a stand-alone product. DB2 is required when you use Tivoli Directory Server or z/OS® LDAP servers as the user registry for Security Access Manager. For z/OS LDAP servers, you must separately purchase DB2.

You can find more information about DB2 at:

http://www.ibm.com/software/data/db2

IBM WebSphere products

The installation packages for WebSphere Application Server Network Deployment, version 8.0, and WebSphere eXtreme Scale, version 8.5.0.1, are included with Security Access Manager version 7.0. WebSphere eXtreme Scale is required only when you use the Session Management Server (SMS) component.

WebSphere Application Server enables the support of the following applications:

- Web Portal Manager interface, which administers Security Access Manager.
- Web Administration Tool, which administers Tivoli Directory Server.
- Common Auditing and Reporting Service, which processes and reports on audit events.
- Session Management Server, which manages shared session in a Web security server environment.
- Attribute Retrieval Service.

You can find more information about WebSphere Application Server at:

http://www.ibm.com/software/webservers/appserv/was/library/

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

Visit the IBM Accessibility Center for more information about IBM's commitment to accessibility.

Technical training

For technical training information, see the following IBM Education website at http://www.ibm.com/software/tivoli/education.

Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

The *IBM Security Access Manager for Web Troubleshooting Guide* provides details about:

- What information to collect before you contact IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The **Community and Support** tab on the product information center can provide more support resources.

Chapter 1. Overview

The IBM Security Web Gateway Appliance is a network appliance-based security solution that provides both access control and protection from web-based threats.

The main features of the appliance include:

- Front-end load balancing capabilities.
- Web Application Firewall capabilities.
- A dashboard for viewing system status such as system notifications, reverse proxy health status, and disk usage.
- Analysis and diagnostics tools such as event logs, memory statistics, and CPU utilization.
- Centralized management of reverse proxy settings such as runtime components, dynamic URL configuration files, and SSL certificates.
- Control of system settings such as updates, licenses, and network settings.

Most of the features are configurable by using either the local management interface or the web services interface.

Appliance format

The IBM Security Web Gateway Appliance comes in two formats: hardware appliance and virtual appliance.

The hardware appliance consists of the appliance hardware and preinstalled Web Gateway Appliance firmware. The hardware appliance has the following machine specifications:

- Intel i7 2600 processor
- 32-GB memory
- 100-GB solid-state drive
- 6 network ports

  Note: Two of these ports are dedicated to the management of the appliance.

The virtual appliance is a Security Access Manager component. It can be hosted by the following virtual hypervisors:

- VMware ESX 4.1 and future fix packs
- VMware ESXi 4.1 and future fix packs
- VMware ESXi 5.0 and future fix packs

Supported Web Reverse Proxy functionality

The IBM Security Web Gateway Appliance Web Reverse Proxy functionality is based on the technology included with the IBM Security Access Manager WebSEAL product. The appliance supports the majority of features that are offered by WebSEAL, with the exception of the items contained in the following table:

Table 1. WebSEAL features that the appliance does not support

Feature	Description
Custom libraries, including CDAS and EAS	The appliance does not support custom CDAS modules. As a result, the appliance does not support the following authentication mechanisms: • IP address • HTTP header • Post password change WebSEAL does not provide CDAS modules for these mechanisms. **Note:** The appliance does support the IBM Security Identity Manager Password Synchronization Plug-in. For more information, see the **[itim]** stanza in the *IBM Security Web Gateway Appliance: Web Reverse Proxy Stanza Reference*.
RSA token	By default, the appliance does not support RSA token authentication. However, you can implement an EAI for token authentication.
Kerberos (Windows Desktop Single Signon)	The appliance does not internally support Kerberos authentication. However, you can configure an EAI to handle Kerberos authentication.
Local junctions	The following limitations apply to local junction support on the appliance: • The appliance can support a single fixed file system path for the local junction of a WebSEAL instance. • Local junctions on the appliance cannot execute any CGI scripts.
Hardware Based Cryptography	The appliance does not support any hardware-based cryptography. However, the hardware appliance does include AES-NI support in the i7-2600 processor, which can handle cryptographic operations.
Application Response Measurement (ARM)	WebSEAL software includes support for ARM to monitor transactions throughout the request and response processing stream. The appliance does not include ARM support.
Tivoli® Common Directory Logging	The Tivoli Common Directory Logging feature stores all log files for IBM Security software applications in a common file system directory. The appliance does not support this common logging. Logging for the appliance is managed through the LMI.
Auditing to a pipe or CARS	The appliance cannot send audit records directly to a pipe or a CARS server. It can however, use an intermediate ISAM authorization server to indirectly send audit records to the destinations.

Table 1. WebSEAL features that the appliance does not support (continued)

Feature	Description
ARS (web service)	The IBM Security Access Manager for Web ARS web service can send request information to an external ARS server for authorization. ARS is not available on the appliance.

Tips on using the appliance

These tips might be useful during the administration of the appliance.

Backup

It is important to back up your appliance frequently. To back up the appliance, use the snapshot facility that is provided by the appliance.

A *snapshot* is a copy of the state of the appliance at a certain time. By using snapshot files, you can back up your appliance and restore the appliance later. It is a good practice to take snapshots regularly and download them from the appliance to serve as a backup. However, snapshots can consume much disk space and as such it is best to clean up the old snapshots regularly.

For details about working with snapshots, see "Managing snapshots" on page 308.

Disk space usage

The disk space in a hardware appliance is limited by the capacity of the installed hard disk. Certain files can use up a significant amount of disk space over time. Such files typically include:

Support files
Support files are used by IBM support personnel to troubleshoot problems with the appliance. The support files contain all log files, temporary and intermediate files, and command output that is needed to diagnose customer support problems. The size of these files can grow large over time. To reduce the disk space that is occupied by these files, download unused support files to an external drive. Then, delete the support files from the appliance. For detailed instructions, see "Managing support files" on page 308.

Snapshot files
Snapshot files record the state that the appliance is in at a certain time. They can be used to restore the appliance to a previous state. The snapshot files are stored on the appliance by default. To reduce the disk space that is used, you can download the snapshot files to an external drive and then delete them from the appliance. For detailed instructions, see "Managing snapshots" on page 308.

Web reverse proxy log files
These log files record the events and activities of the web reverse proxies during the daily operation of the appliance. There are two ways to reduce the disk space that is consumed by these files.

- Configure the web reverse proxy to send the log information to a remote server. For details, see "Logging" on page 110 and "Tracing control" on page 103.

- Clear the unused log files regularly. For details, see "Managing reverse proxy log files" on page 19. Alternatively, use the command-line interface to back up the log files to a USB device, and to purge all log files that were rolled over. For details, see "Archiving and deleting Web Reverse Proxy log files with the command-line interface" on page 23.

  Note: The appliance is configured to automatically purge old log files if the disk utilization reaches a certain threshold (by default, this is configured at 95%). Individual log files are deleted when the disk utilization passes the threshold. The deletion starts with the oldest files until the disk utilization falls below the threshold.

The administrator must monitor the remaining free disk space, and take the necessary actions to ensure that there is adequate disk space. The appliance provides a Disk Usage dashboard widget for administrators to monitor the current disk usage. For more information about managing the disk space, see "Viewing disk usage" on page 14.

Chapter 2. Getting Started

Complete the following tasks that apply to your appliance format.

Hardware appliance tasks

For the hardware appliance, after you determine where to place the IBM Security Web Gateway Appliance in your network, complete the following tasks.

- Install the network cabling.
- Connect to the local management interface (LMI) or a serial console.
- Configure the initial appliance settings.

Connecting cables and starting the appliance

Connect the IBM Security Web Gateway Appliance to your network after you determine where you want to place it on the network.

Procedure

1. Connect the power cable to the IBM Security Web Gateway Appliance.
2. Connect Management Interface 1 to the network you want to use to manage the appliance.
3. Connect the network cables to the application interfaces.
4. Turn on the appliance.

Options to configure the hardware appliance

You can use either a serial console device that is connected to the appliance or the LMI to configure the hardware appliance.

The LMI is the preferable option as it offers more advanced configuration options.

To use a serial console device, you must connect the console device to the hardware appliance with a serial cable. For instructions, see "Connecting a serial console to the appliance."

To use the LMI to configure the appliance, you must browse to the IP address of the appliance. If you do not know the IP address of the appliance, follow instructions in "Determining the system IP address" on page 6.

Connecting a serial console to the appliance

You must connect a serial console to the hardware appliance before you can proceed with the configuration tasks through the command-line interface (CLI).

Procedure

1. Connect the console device to the hardware appliance with a serial cable.
2. If you use a computer as the console device, connect to the appliance with Microsoft Hyperterminal, or another terminal emulation program by using the following settings:

Option	Description
Communication Port	Typically `COM1`

Option	Description
Emulation	VT100
Bits per second	9600
Data bits	8
Parity	None
Stop bits	1
Flow control	None

3. Follow the instructions in "Common tasks" on page 7 to configure initial appliance settings.

Determining the system IP address

If you want to use the LMI to configure the appliance, use one of the following methods to determine the assigned appliance IP address so that you can access the LMI.

- **Method 1:** Use the LCD panel to determine the IP address of the appliance.
  1. Press **OK** on the LCD panel to view the main menu.

     Note: The **OK** button is labeled with an arrow.
  2. Use the arrows to select **IP Address**.
  3. Press **OK**.

     The LCD panel displays the IP address of the IBM Security Web Gateway Appliance. Take note of the address.
- **Method 2:** Use zero-configuration networking to discover the appliance on your network.

  Because the appliance uses a set of industry standard IP protocols, it can be discovered automatically when it is physically connected to your network.

Virtual appliance tasks

For the virtual appliance, connect to the local management interface or the virtual console to configure the initial appliance settings.

Setting up the virtual network

A VMWare environment must be correctly configured before the appliance installation is attempted. The administrator who is installing the appliance must be familiar with VMWare networking concepts.

The virtual appliance installation does not support scripts or a silent mode installation. To install multiple virtual appliances, you can install the first appliance manually and then use VMware ESX or vSphere to make copies of the virtual machine.

Installing the virtual appliance using VMware

Use the provided .iso image to install the virtual appliance.

Procedure

1. Create a new virtual machine with your VMware ESX or vSphere.

   Note:

- The instructions for creating a virtual machine might differ depending on your VMware ESX or vSphere version. See the VMware documentation that suits your version for specific instructions.
- Ensure that the virtual machine has enough disk space that is allocated to store the configuration and log file data for the appliance. Allocate at least 100 GB of disk space for the appliance.
- Specify **Virtual Machine Version: 7** as your virtual machine version.
- Specify **Linux** as the guest operating system and **Other 2.6x Linux (64-bit)** as the guest operating system version.
- The memory size has influence over how many WebSEAL instances can be created and how many sessions can be active at a single point in time. The minimum recommended memory size is 4096 MB.
- The appliance needs four to six virtual network adapters. The first two network adapters serve as management interfaces. The rest of the network adapters serve as application interfaces. Four network adapters can be configured using the wizard. Additional network adapters can be added upon completion of the wizard. Each network adapter must be of the type **E1000**.)
- For SCSI controller, select **LSI Logic Parallel**.
- For Virtual Device Node, select **SCSI (0:0)**.

2. Configure the virtual machine to boot from the supplied .iso file and then start the virtual machine. The installer executes automatically.
3. Select the language to be used during the installation by entering the number that corresponds to the desired language.
4. Enter `YES` to proceed with the installation. Alternatively if you do not want to proceed with the installation, enter `NO` to move to the reboot prompt.
5. Examine the installation messages to ensure that the installation was successful. After the installation process has completed, unmount the installation media and then press the **Enter** key to reboot the appliance.
6. When the reboot operation has finished, you can start the console-based appliance setup wizard by logging on as the `admin` user with a password of `admin`. Alternatively, the Appliance Setup wizard can be accessed through the LMI.

Common tasks

These tasks are common for both the hardware appliance and the virtual appliance.

You can choose either of the following methods to configure initial appliance settings.

- Command-line interface (CLI)
- Local management interface (LMI)

The LMI method offers more advanced configuration options.

Command-line interface initial appliance settings wizard

The initial appliance settings wizard runs the first time that an administrator logs in to the command-line interface (CLI) of an unconfigured appliance.

Navigation

You can move between screens in the wizard using the following options:

- p: Previous Screen
- n: Next Screen

To cancel the setup process at any time, use the exit command.

Modules

You must configure the following modules to set up your appliance:

Module	Description
Welcome	Describes the appliance settings that you can configure using the wizard.
Software License Agreement	Describes the appliance license agreement, IBM terms, and non-IBM terms.
Password Configuration	Changes your password.
Host Configuration	Changes the host name.
Management Interface Settings	Configures the management network interfaces. Displays device settings and the current working-set policy for the primary and secondary interfaces.
DNS Configuration	Configures the DNS servers that are used by the appliance.
Time Configuration	Configures the time, date, and time zone on the appliance.

Local management interface Appliance Setup wizard

The Appliance Setup wizard runs the first time that an administrator logs in to the local management interface (LMI) of an unconfigured appliance.

After you log in to the LMI for the first time, follow the Appliance Setup wizard to complete the initial configuration of the appliance. The tasks that you must complete for the initial configuration include:

- Read and accept the License Agreement.
- Download and install the license file. You must install the license to download the firmware for the hardware appliance and IBM X-Force updates.
- Set the appliance password.
- Configure the networking, which includes the host name, management interface settings, and DNS configuration.
- Configure the application interface settings.
- Configure the date and time settings.

When you complete the basic configuration, a summary screen displays. Review the details on the completion page and click **Complete Setup**.

Chapter 3. Managing the appliance

The appliance provides three mechanisms by which it can be managed: the local management interface (LMI), the command-line interface (CLI), and web services interface.

Local management interface

The IBM Security Web Gateway Appliance offers a browser-based graphical user interface for local, single appliance management.

The following paragraphs are general notes about the usage of the local management interface (LMI). Examples of specific commands using the LMI are provided through the remainder of this document.

To log in to the LMI, type the IP address or host name of your appliance into your web browser. The following web browsers are supported:

- Windows
  - Google Chrome, version 19.0 or later
  - Microsoft Internet Explorer, version 9 or later
  - Mozilla Firefox, version 12.0 or later
- Linux/AIX®/Solaris
  - Mozilla Firefox, version 12 or later

Use the default credentials to log in to the local management interface for the first time:

- **User Name:** admin
- **Password:** admin

After you log in for the first time, use the first-time configuration pages to change your password.

To log out of the local management interface, click **Logout**.

Command-line interface

Access the command-line interface (CLI) of the appliance by using either an ssh session or the console.

The following paragraphs are general notes about the usage of the CLI. Examples of specific commands using the CLI are provided through the remainder of this document.

The following example shows the transcript of using an ssh session to access the appliance:

```
usernameA@example.ibm.com>ssh -l admin webapp.vwasp.gc.au.ibm.com
admin@webapp.vmasp.gc.au.ibm.com's password:
Welcome to the IBM Security Web Gateway
Enter "help" for a list of available commands
webapp.vwasp.gc.au.ibm.com>
```

The method to access the console differs between the hardware appliance and the virtual appliance:

- For the hardware appliance, a serial console device must be used. For more information about attaching a serial console device to the hardware, see "Connecting a serial console to the appliance" on page 5.
- For the virtual appliance, you can access the console by using the appropriate VMWare software.

  For example, VMWare vSphere Client.

Note: The CLI contains only a subset of the functionality available from the local management interface. The following list gives a high-level overview of the functions available from the command-line interface:

- Work with firmware images.
- Work with fix packs.
- Work with hardware settings.
- Work with licenses.
- Work with the local management interface.
- Work with management settings.
- Work with policy snapshot files.
- Work with support information files.
- Work with network diagnostic tools.
- Work with firmware and security updates.
- Work with the Web Gateway settings.

Web service

The appliance can also be managed by sending RESTful web service requests to the appliance.

The following paragraphs are general notes about the usage of the web service interface. The content and format of these web service requests are explained through the remainder of this document.

Required header for calling a web service

All web service requests must include the following two headers.

Accept:application/json
: The `accept` header must be present and the value must be set to `application/json`. If the header is missing, or set as a different value, the web service request fails.

BA header
: Each request must contain a BA header with a valid user name and password. If this header is missing, the request fails.

The following example is the valid request format for retrieving the list of reverse proxy instances by using curl.

```
curl -k -H "Accept:application/json" --user username:password
https://{appliance_hostname}/reverseproxy
```

Note: The previous list contains only two headers that are mandatory for all web service requests. It is not an extensive list of headers that are required for all request actions. The previous example shows a curl GET request on a resource

URI. This request requires only the two mandatory headers that are listed. Other HTTP methods, such as POST or PUT, require more headers. The following example is a valid request for starting a reverse proxy instance called `inst1` using curl:

```
curl -k -H "Accept:application/json" -H "Content-type:application/json"
--user username:password --data-binary '{ 'operation':'start' }'
-X PUT https://{appliance_hostname}/reverseproxy/inst1
```

Notice the additional required header **Content-type** for the PUT operation.

Other HTTP clients, such as Java, might require more headers. For required headers for RESTful web services, check the HTTP client documentation.

Web service responses

The response to a web service call is composed of two components: HTTP response code and JSON message.

The response to a successful web service request includes a 200 status code, and JSON data that contains context-specific information about the request processing. The response to an unsuccessful web service request includes an HTTP error response code, and JSON data that contains the error message.

HTTP response codes

Table 2. HTTP error response codes

Code	Description
200	Success.
400	There is a problem with the request. The JSON message describes the problem.
404	The resource that is specified in the request does not exist. The JSON message indicates which resource.
500	An internal error was encountered while the request is processed. The JSON message indicates the problem.

JSON error response format

```
{"message":"The error message"}
```

Chapter 4. Home: Appliance Dashboard

The appliance provides a serious of dashboard widgets in its local management interface. You can use these widgets to view commonly used system information.

These widgets are displayed right after you log in. You can also access them by clicking **Home: Appliance Dashboard** on the menu bar.

Viewing system notifications

You can view warning information about potential problems with the Notification dashboard widget.

Procedure

1. From the dashboard, locate the Notification widget. Warning messages about the following potential problems are displayed:
   - Certificates that are due to expire.
   - Reverse proxy instances that are not currently running.
   - The disk space utilization has exceeded the warning threshold.
   - The CPU utilization has exceeded the warning threshold.
2. Take appropriate actions as required.

Viewing reverse proxy health status

The health status of a reverse proxy is determined by the state of instances, junctions, and junctioned servers. You can view the health status information with the Reverse Proxy Health dashboard widget.

Procedure

1. From the dashboard, locate the Reverse Proxy Health widget.

   The health status of each instance, its junctions, and the junctioned servers are displayed in a hierarchical structure. Health status is determined by the health of all elements lower than the current element in the hierarchy.
   - An instance is unhealthy if it is stopped or pdadmin cannot contact it.
   - A junction is unhealthy if it is disabled or pdadmin cannot return information for it.
   - A junctioned server is unhealthy if it is disabled or offline.

   Each element can be in one of the three health states:

Icon	State	Description
	Healthy	All child elements are healthy.
	Warning	The element contains at least one unhealthy child element and at least one healthy child element.
	Unhealthy	All child elements are unhealthy.

2. *Optional:* Click **Refresh** to refresh the health data.

Viewing disk usage

You can view the disk space status and remaining disk life information with the Disk Usage dashboard widget.

Procedure

1. From the dashboard, locate the Disk Usage widget.

   Disk Space Pie Chart
   Information about used disk space and free disk space is visualized in the pie chart.

   Consumed Disk Space
   How much space (in GB) is already used.

   Note: Most of the disk space is typically used by log files and trace files. To minimize the disk footprint, set the appliance to store log and trace files on a remote server. It is also a good practice to clear unused log and trace files on a periodic basis. For details, see "Logging" on page 110 and "Tracing control" on page 103.

   Free Disk Space
   How much space (in GB) is free.

   Total Disk Space
   How much space in total (in GB) is available to the appliance.

   Note: The disk space in a hardware appliance is limited by the capacity of the hard disk drive it carries.

2. *Optional:* Click **Refresh** to refresh the data.

Viewing IP addresses

You can view a categorized list of IP addresses that the appliance is listening on with the Interfaces dashboard widget.

Procedure

1. From the dashboard, locate the Interfaces widget. The IP addresses of all enabled and configured interfaces are displayed, along with the virtual IP addresses that are managed by the front-end load balancer.

   Management IPs
   A list of IP addresses of the management interfaces (M.1, M.2) that are enabled and configured.

   Application IPs
   A list of IP addresses of the application interfaces (P.1, P.2, P.3, P.4) that are enabled and configured.

   Load Balancer IPs
   A list of IP addresses of the load balancer services.

2. *Optional:* Click **Refresh** to refresh the data.

Viewing front-end load balancer health status

The health status of a front-end load balancer is determined by the state of the load balanced servers. You can view the health status information with the Load Balancer Health dashboard widget.

Procedure

1. From the dashboard, locate the Load Balancer Health widget.
   - Under **High Availability** (if high availability is configured):
     - The first row displays the health status of the self front-end load balancer and whether it is active or passive.
     - The second row displays the health status of the peer front-end load balancer and whether it is active or passive.
   - Under **Services** (if at least one service is configured):
     - The health status of the configured services and the load balanced servers are displayed in a hierarchical structure. You can expand a service to view the health status of the servers that are attached to this service.

   Each element can be in one of the following health states:

Icon	State	Description
	Healthy	All child elements are healthy.
	Warning	The element contains at least one unhealthy child element and at least one healthy child element.
	Unhealthy	All child elements are unhealthy.

2. *Optional:* Click **Refresh** to refresh the health data.

Viewing average response time statistics

The Web Reverse Proxy can be configured to record transaction logs. One of the attributes that are recorded is the average request response time. This information is recorded at a per-junction level. To view a summary of the average response time that has been recorded, use the Average Response Time widget.

Procedure

1. From the dashboard, locate the Average Response Time widget. The average response time for requests is displayed on a graph.

   Note: The widget is only displayed if one or more Reverse Proxy instances have the Flow Data function enabled.
2. Under **Reverse Proxy Instances**, select the instance to view the average response time statistics for.
3. Under **Junctions**, select the junctions to display on the graph. Each junction is represented by a separate line on the graph.
4. Under **Date Range**, select the duration over which the response times are recorded.

Viewing security action statistics

The Web Reverse Proxy can be configured to perform inspections on web content, searching for potential malicious requests (known as issues). It can then take certain defensive actions against any discovered issues. A summary of the defensive actions that have been taken can be viewed by using the Security Actions widget.

Procedure

1. From the dashboard, locate the Security Actions widget. The number of times each defensive action has been taken is displayed in a graph.

   Note: The widget is only displayed if one or more instances have the security statistics function enabled.
2. Under **Reverse Proxy Instances**, select the instances to view action statistics for.

   Note: Only instances that have security statistics function enabled are listed for selection.
3. Under **Actions**, select the actions to be included in the statistics. The number of actions that are displayed is the total of all selected actions.
4. Under **Date Range**, select the duration over which the actions are taken.

Viewing certificate expiry

You can view certificate details with the Certificate Expiry widget.

Procedure

1. From the dashboard, locate the Certificate Expiry widget. Details about the certificates are displayed.

   Certificate Label
   Label of the certificate.

   Expiration
   The date on which the certificate expires.

   Type Type of the certificate.

   Key Database
   Name of the key database that the certificate belongs to.
2. *Optional:* Click **Refresh** to refresh the data.

Viewing partition information

You can view information about the active and backup partitions with the Partition Information widget.

Procedure

1. From the dashboard, locate the Partition Information widget. Details about the active and backup partition are displayed.

   Firmware Version
   Version information of the appliance firmware

   Installation Date
   Date on which the appliance firmware was installed

   Installation Type
   Type of the appliance firmware installation

   Last Boot
   Time when the appliance was last booted
2. *Optional:* Click **Firmware Settings** to go the page to modify settings of the firmware.

Viewing reverse proxy throughput

You can view flow data at an appliance-wide level with the Reverse Proxy Throughput widget.

Procedure

1. From the dashboard, locate the Reverser Proxy Throughput widget.
2. Select the duration over which data is collected and displayed with the **Data Range** list.

   By default, data of all configured WebSEAL instances on this appliance from the last 24 hours are displayed.
3. *Optional:* Click **Refresh** to refresh the data.

Viewing network traffic

You can view network traffic for the past hour with the Network Traffic widget.

Procedure

1. From the dashboard, locate the Network Traffic widget. The **In** and **Out** traffic details for the past hour are displayed.
2. *Optional:* Click **P.1**, **P.2**, **P.3**, or **P.4** to display the details for a specific interface.

Chapter 5. Monitoring: Analysis and Diagnostics

You can monitor the health and statistics in the Security Web Gateway Appliance.

Viewing the event log

System events are logged when the system settings are changed and when problems occur with the system. Use the Event Log management page to view system events.

Procedure

1. Click **Monitor Analysis and Diagonostics** > **Logs** > **Event Log**. The system events displayed.
2. Click **Pause Live Streaming** to stop the live updating of the event log.
3. Click **Start Live Streaming** to resume live updating of the event log.

Managing reverse proxy log files

Use the Manage Reverse Proxy Log Files management page to work with reverse proxy log files.

Procedure

1. From the top menu, select **Monitor Analysis and Diagnostics** > **Logs** > **Manage Reverse Proxy Log Files**. Details of all common log files are displayed under **Log Files for Selected Instance**.

   You can use the filter bar under **Name** to filter entries that meet specific conditions. Click **Clear filter** to return to the full list.
2. *Optional:* If instance-specific log files are of interest, select the instance from the list under **Reverse Proxy Instances**. Details of all common log files and instance-specific log files are displayed under **Log Files for Selected Instance**.
3. Work with the reverse proxy log files.
   - **View the content of a reverse proxy log file**
     a. Select the log file that you want to view.
     b. Click **View**. The content of the log file is displayed. By default, the last 100 lines of a log file are displayed if the file is longer than 100 lines. You can define the number of lines to display by entering the number in the **Number of lines to view** field and then click **Reload**. Optionally, you can provide a value in the **Starting from line** field to define the start of the lines. If the **Starting from line** field is set, then the **Number of lines to view** field determines how many lines to view forward from the starting line. If the **Starting from line** field is not set, then the **Number of lines to view** field determines how many lines to view from the end of the log file.

        Note: The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
     c. *Optional:* Click **Export** to download the log file.

Note: You must configure the software to block pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.

- **Export a reverse proxy log file**
  a. Select the log file that you want to export.
  b. Click **Manage** > **Export**.

     Note: You must configure the software to block pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.

  c. Confirm the save operation in the browser window to export the file to a local location.
- **Clear a reverse proxy log file**
  a. Select the log file that you want to clear.
  b. Click **Clear**.
  c. On the Confirm Action confirmation page, click **Yes**.

Viewing memory statistics

View the memory graph to see the memory utilization of the Security Web Gateway Appliance.

Procedure

1. Click **Monitor Analysis and Diagnostics** > **System Graphs** > **Memory**.
2. Select a **Date Range**:

Option	Description
1 Day	Displays data points for every minute during the last 24 hours.
3 Days	Displays data points for every 5 minutes during the last three days. Each data point is an average of the activity that occurred in that hour.
7 Days	Displays data points every 20 minutes during the last seven days. Each data point is an average of the activity that occurred in that hour.
30 Days	Displays data points for every hour during the last 30 days. Each data point is an average of the activity that occurred in that hour.

3. In the Legend box, select **Memory Used** to review total memory utilization.

Viewing CPU utilization

View the CPU graph to see the CPU utilization of the Security Web Gateway Appliance.

Procedure

1. Click **Monitor Analysis and Diagnostics** > **System Graphs** > **CPU**.
2. Select a **Date Range**:

Option	Description
1 Day	Displays data points for every minute during the last 24 hours.
3 Days	Displays data points for every 5 minutes during the last three days. Each data point is an average of the activity that occurred in that hour.
7 Days	Displays data points every 20 minutes during the last seven days. Each data point is an average of the activity that occurred in that hour.
30 Days	Displays data points for every hour during the last 30 days. Each data point is an average of the activity that occurred in that hour.

3. In the Legend box, select the CPU utilization data that you want to review:
   - User
   - System
   - Idle

Viewing storage utilization

View the storage graph to see the percentage of disk space that is used by the boot and root partitions of the Security Web Gateway Appliance.

Procedure

1. Click **Monitor Analysis and Diagnostics** > **System Graphs** > **Storage**.
2. Select a **Date Range**:

Option	Description
1 Day	Displays data points for every minute during the last 24 hours.
3 Days	Displays data points for every 5 minutes during the last three days. Each data point is an average of the activity that occurred in that hour.
7 Days	Displays data points every 20 minutes during the last seven days. Each data point is an average of the activity that occurred in that hour.
30 Days	Displays data points for every hour during the last 30 days. Each data point is an average of the activity that occurred in that hour.

3. In the Legend box, select which partitions you want to review:

   Boot The boot partition.

   Root The base file system, where the system user is root.

Viewing application interface statistics

To view the bandwidth and frames that are being used on your application interfaces, use the Application Interface Statistics management page.

Procedure

1. From the top menu, select **Monitor Analysis and Diagnostics** > **Network Graphs** > **Application Interface Statistics**.
2. In the **Date Range** field, select the period to display the statistics for.

Option	Description
1 Day	Displays data for every 20-minute interval in one day.
3 Days	Displays data for every 20-minute interval during the last three days.
7 Days	Displays data for every 20-minute interval during the last seven days.
30 Days	Displays data for every day during the last 30 days.

Viewing reverse proxy traffic

To view flow data at an instance-specific level with the local management interface, use the Reverse Proxy Traffic management page.

Procedure

1. From the top menu, select **Monitor Analysis and Diagnostics** > **Reverse Proxy Graphs** > **Reverse Proxy Traffic**.
2. On the Reverse Proxy Traffic page, specify the settings for the chart displayed.

   Instance
   The instance which the data displayed are specific to.

   Aspect Type
   The type of chart to display the data with. Select one from **Column and Lines**, **Column**, and **Lines**.

   Start Date
   The starting date.

   Start Time
   The starting time of the day.

   Date Range
   The duration over which data is collected and displayed. Select from **1 Hour** to **30 Days**.

   For example, if the date and time that is chosen is `04.12.2012 10.00` and the duration is `12 Hours`, the data that are collected between 10:00 a.m. and 10:00 p.m. on 12th April 2012 are displayed.

   By default, data of the first instance in the instance list for the last 24 hours are displayed, grouped by junction.

Viewing reverse proxy throughput

To view flow data at an appliance-wide level with the local management interface, use the Reverse Proxy Throughput management page.

Procedure

1. From the top menu, select **Monitor Analysis and Diagnostics** > **Reverse Proxy Graphs** > **Reverse Proxy Throughput**.
2. On the Reverse Proxy Throughput page, specify the settings for the chart displayed.

   Chart Type
   The type of chart to display the data with. Select one from **Column and Lines**, **Column**, and **Lines**.

   Date Range
   The duration over which data is collected and displayed. Select from **1 Hour** to **30 Days**.

   Start Date
   The starting date.

   Start Time
   The starting time of the day.

   For example, if the date and time that is chosen is `04.12.2012 10.00` and the duration is `12 Hours`, the data that are collected between 10:00 a.m. and 10:00 p.m. on 12 April 2012 are displayed.

   By default, data of all configured WebSEAL instances on this appliance from the last 24 hours are displayed.

Archiving and deleting Web Reverse Proxy log files with the command-line interface

Use the `logs` option in the command-line interface to archive Web Reverse Proxy log files to a USB device and then delete old log files to free up disk space.

Procedure

1. In the command-line interface, go to **wga** > **logs**.
2. *Optional:* Enter `help` to display all available commands.

```
Current mode commands:
archive         Archive the log files to a USB device.
delete          Delete the log files which have been rolled over by the system.
Global commands:
back            Return to the previous command mode.
exit            Log off from the appliance.
help            Display information for using the specified command.
reboot          Reboot the appliance.
shutdown        End system operation and turn off the power.
top             Return to the top level.
```

3. Archive or delete the log files.
   - **Archive the log files to a USB device**
     a. Enter `archive` to save the log files to a USB device.
     b. Insert a USB device into the USB port of the appliance.
     c. Enter `YES` to start the archive operation. A list of archived files are displayed, along with a message that indicates when the archive operation has completed. Example output is shown as follows:

```
updating: var/PolicyDirector/log/ (stored 0%)
updating: var/PolicyDirector/log/msg__pdmgrd_utf8.log (deflated 85%)
updating: var/PolicyDirector/log/PDMgr_config_start.log (deflated 37%)
updating: var/PolicyDirector/log/ivmgrd.pid (stored 0%)
updating: var/pdweb/default/log/ (stored 0%)
updating: var/pdweb/default/log/iss-pam1.so (deflated 59%)
updating: var/pdweb/default/log/webseald-default.pid (stored 0%)
updating: var/pdweb/default/log/config_data__default
-webseald-felbb.wga.gc.au.ibm.com.log (deflated 92%)
updating: var/pdweb/default/log/referer.log (stored 0%)
updating: var/pdweb/default/log/msg__webseald-default.log (deflated 89%)
updating: var/pdweb/default/log/pam.log (deflated 98%)
updating: var/pdweb/default/log/agent.log (stored 0%)
updating: var/pdweb/default/log/request.log (stored 0%)
The log files have been successfully archived to the USB drive:
iswga_logs.zip. It is now safe to remove the USB drive.
```

d. Remove the USB device from the USB port.

- **Delete the log files**
  a. Enter `delete` to purge all log files that are rolled over.
  b. Enter `YES` to confirm the delete operation.

Chapter 6. Secure: Reverse Proxy Settings

You can use the appliance to manage your reverse proxy settings.

Some of the settings apply globally to all reverse proxy instances. Others are specific to certain instances. See details in the following topics.

Migration

You can migrate a WebSEAL instance that is running as software to the appliance.

Note: Each environment is different. Although tools are provided to assist with the migration of WebSEAL instances, the administrator is responsible to understand the migration process and configure the tools accordingly.

The high-level steps for the migration include:

Preparation

1. Custom CDAS or EAS libraries are not supported. Make sure that there is no dependency on custom CDAS or EAS libraries before you start to migrate the system. For example, any custom CDAS processing must be converted to an EAI.
2. Local junctions are supported, but a fixed location is used as the document root. A local junction is also not permitted to run any CGI scripts. It can serve only static page content. Any CGI scripts must be migrated to a remote server. The appliance supports only a single local junction. The content for all other local junctions (if any) must also be migrated to a remote server.
3. On the source WebSEAL server, create the directory structure of configuration files, as defined in the following table. Only those directories for which files are migrated must be created. Create these directories as subdirectories under a single source directory.

Table 3. Directory structure

Directory	Directory
dynurl	Dynamic URL configuration files.
fsso	Forms-Based Single Sign-on configuration files.
jmt	Junction Mapping Table configuration files.
keytab	The key database (kdb/sth) files which are used by the WebSEAL instance. This does not include the keyfile which is used to communicate with the policy server.
ltpa-keys	LTPA key files.
tam-keys	Key files that are generated with the **cdsso-key-gen** utility. They are used for things such as encrypting the failover cookie.
xslt/user-map-cdas	XSLT configuration file that is used by the client certificate user mapping CDAS.
xslt/http-transformation	XSLT configuration file that is used by the HTTP transformation rules function.
doc-root/docs	The files that are served by the WebSEAL local junction. These files are typically located under the `/opt/pdweb/www-<instance>/lib/docs` directory.

Table 3. Directory structure (continued)

Directory	Directory
doc-root/errors	The error pages that are served by the WebSEAL instance. These files are typically located under the /opt/pdweb/www-<*instance*>/lib/errors directory.
doc-root/html	The management HTML pages (for example, login.html) which are served by the WebSEAL instance. These files are typically located under the /opt/pdweb/www-<*instance*>/lib/html directory.
doc-root/oauth	The OAuth response files, as defined within the [oauth-eas] stanza of the WebSEAL configuration file.
junctions	The XML files that contain the junction definitions for the WebSEAL instance. These files are typically located under the /opt/pdweb/www-<*instance*>/jct directory.
etc	The configuration files that are used by the WebSEAL instance. In particular, the routing file, the webseald-<instance>.conf and the webseald-<instance>.conf.obf files.

Note: When you create the directory structure, additional subdirectories are not supported for any directory other than the doc-root ones (doc-root/docs, doc-root/errors, doc-root/html, doc-root/oauth). For example, you can create a directory structure such as /doc-root/error/<folder>/<file>, but a structure such as xslt/http-transformation/<folder>/<file> is not valid. For directories other than the doc-root ones, files can be placed only in the default main directories that are listed in Table 3 on page 25. For example, xslt/http-transformation/<file>.

Note: All files to be copied must have unique file names. If two files have the same name, the migration tool copies only the first file that matches the name. For example, in the following structure:

```
[http-transformation]
request_pop1 = <path1>/pop1.xsl
response_pop1 = <path2>/pop1.xsl
```

Only <path1>/pop1.xsl are created in the directory structure. All references to <path1>/pop1.xsl and <path2>/pop1.xsl in the configuration file are reduced to pop1.xsl, which now points to the same file.

Perl utility for collecting files

A Perl utility is provided to help facilitate the collection of files that are required by the WebSEAL instance. This utility can process the configuration for the specified WebSEAL instance. It can also copy the necessary files into the directory structure that is required by the import facility of the Web Gateway appliance.

To set up and run this utility, follow these steps:

a. Obtain the Perl script by downloading it from the appliance at the following URL:

   https://<appliance>/reverseproxy/wga_migrate.pl

b. Copy the script to the WebSEAL server.

c. Ensure that Perl is installed and available on the WebSEAL server.

d. Locate the name of the configuration file for the WebSEAL instance that is to be migrated.

e. Run the `wga_migrate.pl` script, specifying the name of the WebSEAL configuration file and the destination directory. The format for using the script is as follows:

```
perl wga_migrate.pl [-c config-file] [-d dst-dir] {-v}
```

-c *config-file*	The name of the WebSEAL configuration file.
-d *dst-dir*	The name of the destination directory. This directory must not exist on the file system.
-v	Display more status messages during the execution of the script.

For example:

```
perl wga_migrate.pl -c /var/pdweb/etc/webseald-default.conf
-d /tmp/migrate_out
```

f. Review the files that are contained within the destination directory to ensure that all of the necessary files have been located.

4. The WebSEAL configuration file must be included in the set of configuration files to be migrated. The obfuscated configuration file, as defined by the `[configuration-database]` stanza and **`file`** configuration entry, must also be included. Do not change the name of the obfuscated database (`webseald-<instance>.conf.obf`). If the obfuscated database file name is different, rename it back to the default and update the configuration entry accordingly.
5. Modify the copied WebSEAL configuration file so that any configuration entries that are not applicable to the new WebSEAL instance are removed. Examples of entries that you might potentially not want to migrate would include network settings. The following configuration entries are ignored when the configuration file is imported into the appliance:
   - **`token-card`** configuration entry from the `[authentication-levels]` stanza
   - **`server-name`** configuration entry from the `[server]` stanza
   - **`network-interface`** configuration entry from the `[server]` stanza
   - `[interfaces]` configuration stanza
6. Create a compressed file, with the contents being relative to the location that contains the copied files. For example, if the directory structure was created in `/tmp/migrate`, the command would be:

```
cd /tmp/migrate; zip -r /tmp/migrate.zip *
```

Migration

When the migration compressed file is available, you can start the migration.

1. Create a new WebSEAL instance on the appliance with the local management interface. The name of this instance must match the name of the instance to be migrated.
2. Import the migration compressed file.

   Note: If you are warned that files might be overwritten as a part of the import operation, you must validate the overwrite operation before continuing. Make sure that the overwrite operation does not affect any other WebSEAL instances that might be running on the appliance.

   - For detailed steps when you import with the local management interface, see Import the contents of a zip file into the administration pages root.

- For detailed steps when you import with the web service, see "Importing the contents of a .zip file to the administration pages root with web service" on page 96.

3. Deploy the changes.
4. Restart the WebSEAL instance.
5. Examine the WebSEAL log file for any potential migration issues.

Configuration changes commit process

The LMI uses a two-stage commit process when you make changes to the appliance.

Stage 1
Changes are made by using the LMI and saved to a staging area.

Stage 2
The user explicitly deploys the changes into production.

Multiple changes can exist in a pending state at the same time. They are committed or rolled back together when a user deploys or rolls back these changes.

Any changes that affect running reverse proxy instances require a restart of the effected instances before the changes can take effect.

Certain appliance updates require either the appliance or the web server to be restarted before the changes can take effect. When one or more of these updates are made alongside other reverse proxy updates, an additional step is required to deploy the reverse proxy updates. You must:

1. Deploy all updates.
2. Restart the appliance or the web server.
3. Deploy all remaining updates.

If there are conflicts between the pending changes and the production files, then all pending changes are automatically rolled back and the production files remain unchanged.

Web service

Deploy the pending configuration changes

URL
`https://{appliance_hostname}/pending_changes/deploy`

Method
`GET`

Parameters
N/A

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/pending_changes/deploy
```

Response:

```
200 ok
```

Roll back the pending configuration changes

URL

```
https://{appliance_hostname}/pending_changes/forget
```

Method

```
GET
```

Parameters

N/A

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/pending_changes/forget
```

Response:

```
200 ok
```

Retrieve the number of outstanding changes

URL

```
https://{appliance_hostname}/pending_changes/count
```

Method

```
GET
```

Parameters

N/A

Response

HTTP response code and JSON data that represents the number of pending changes.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/pending_changes/count
```

Response:

```
{"count": 3}
```

Retrieve the list of outstanding changes

URL

```
https://{appliance_hostname}/pending_changes
```

Method

GET

Parameters

N/A

Response

HTTP response code and JSON data that represents the list of pending changes.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/pending_changes
```

Response:

```
200 ok

[{
"id": 0,
"policy": "SSL Certificates",
"user": "admin",
"date": "2012-11-05T11:22:20+10:00"
}]
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

local management interface

When there are pending changes, a warning message is displayed at the top of the main pane. To deploy or roll back the pending changes:

1. Click the **Click here to review the changes or apply them to the system** link within the warning message.
2. In the Deploy Pending Changes page:
   - To view the details of changes that are made to a particular module, click the link to that module.
   - To deploy the changes, click **Deploy**.
   - To abandon the changes, click **Roll Back**.
   - To close the pop-up page without any actions against the changes, click **Cancel**.

Runtime components

Use either the local management interface or web service to manage and configure the runtime environment.

Managing runtime components with local management interface

In the local management interface, go to **Secure Reverse Proxy SettingsManageRuntime Component**.

Viewing the runtime environment configuration status with local management interface

To view the runtime environment configuration status with the local management interface, use the Runtime Component management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Runtime Component**.
2. The configuration status of the runtime environment is displayed.

Managing runtime configuration files with local management interface

To manage configuration files with the local management interface, use the Runtime Component management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Runtime Component**.
2. Click **Edit**.
3. Select the runtime configuration file of interest.
4. Edit the configuration file and then click **Save** to save the changes. If you do not want to save the changes, click **Cancel**. If you want to revert to the previous version of the configuration file, click **Revert**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Configure the runtime environment with local management interface

To configure the runtime environment with the local management interface, use the Runtime Component management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Runtime Component**.
2. Click **Configure**. You can configure your policy server to be local or remote.
   - **Local policy server with a remote LDAP user registry**
     a. Under **Policy Server**, select **Local**.
     b. Under **User Registry**, select **LDAP Remote**.
     c. Click **Next**.
     d. On the Policy Server tab, provide settings for the fields displayed. Fields with an asterisk are required and must be completed.
        - **Management Suffix**: The LDAP suffix that is used to hold the IBM Security Access Manager secAuthority data.
        - **Management Domain**: The IBM Security Access Manager domain name.
        - **Administrator Password**: The security administrator's password.
        - **Confirm Administrator Password**: The security administrator's password.
        - **SSL Server Certificate Lifetime (days)**: The lifetime in days for the SSL server certificate.

- **SSL Compliance**: Specifies any additional SSL compliance.

e. Click **Next**.

f. On the LDAP tab, provide settings for the fields displayed.
   - **Host name**: The name of the LDAP server.
   - **Port**: The port to be used the system communicates with the LDAP server.
   - **DN**: The distinguished name that is used when the system contacts the user registry.
   - **Password**: The password for the DN.
   - **Enable SSL**: Whether SSL is enabled.
   - **Certificate Database**: The KDB file that contains the certificate that is used to communicate with the user registry. This field is required if "Enable SSL" is selected.
   - **Certificate Label**: The label of the SSL certificate that is presented to the user registry upon request. This field is optional and is only required if SSL is enabled, and the user registry is configured to require a client certificate.

g. Click **Finish** to save the settings.

- **Local policy server with a local user registry**

  Note: Users and groups within the local user registry are managed through the Security Access Manager administration framework; for example, pdadmin. All these users and groups are housed under the suffix "dc=iswga".

  a. Under **Policy Server**, select **Local**.

  b. Under **User Registry**, select **LDAP Local**.

  c. Click **Next**.

  d. On the Policy Server tab, provide settings for the fields displayed. Fields with an asterisk are required and must be completed.
     - **Administrator Password**: The security administrator's password.
     - **Confirm Administrator Password**: The security administrator's password.
     - **SSL Server Certificate Lifetime (days)**: The lifetime in days for the SSL server certificate.
     - **SSL Compliance**: Specifies any additional SSL compliance.

  e. Click **Finish** to save the settings.

- **Remote policy server**

  a. Under **Policy Server**, select **Remote**.

  b. Under **User Registry**, select whether to use **LDAP** or **Active Directory**.

  c. Click **Next**.

  d. On the Policy Server tab, provide settings for the fields displayed.
     - **Host name**: The name of the host that hosts the IBM Security Access Manager policy server.
     - **Port**: The port over which communication with the IBM Security Access Manager policy server takes place.
     - **Management Domain**: The IBM Security Access Manager domain name.

  e. Click **Next**.

– If **LDAP** is selected as the user registry in previous steps, complete settings on the **LDAP** tab.
  - **Host name**: The name of the LDAP server.
  - **Port**: The port to be used when the system communicates with the LDAP server.
– If **Active Directory** is selected as the user registry in previous steps, complete settings on the **Active Directory** and **Active Directory SSL** tabs.
  1) On the **Active Directory** tab, provide settings for the following items:
     - **Host name**: The name of the Active Directory server.
     - **Domain**: The name of the active directory domain.
     - **Configure IBM Security Access Manager in Active Directory multiple domain environment**: Whether the environment is configured against a multi-domain active directory.
     - **Distinguished name (DN) to store IBM Security Access Manager data**: The DN within active directory that is used to store the Security Access Manager data.
     - **Enable the use of an e-mail address as a user ID**: Whether to use an email address as the Security Access Manager user ID.
     - **Global Catalog server host name**: The name of the Active Directory global catalog server.
  2) Click **Next**.
  3) On the **Active Directory SSL** tab, provide settings for the following items:
     - **Enable SSL**: Whether to communicate to the active directory server over SSL.
     - **SSL key file**: The KDB file that contains the certificate that is used to communicate with the user registry. This field is required if "Enable SSL" is selected.
     - **Certificate Label**: The label of the SSL certificate that is used when the system communicates with the user registry. This field is optional and only valid if an SSL key file is entered.

f. Click **Finish** to save the settings.

Unconfiguring the runtime environment with local management interface

To unconfigure the runtime environment component of the appliance with the local management interface, use the Runtime Component management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Runtime Component**.
2. Click **Unconfigure**.
3. Take one of the following sets of actions.
   - **Unconfigure a local policy server with a remote LDAP user registry**
     a. Enter the LDAP DN and LDAP password.
     b. Select the **Clear user registry entries** check box if you want the unconfigure operation to remove all Security Access Manager domain, user, and group information. By default, this check box is not selected.

c. Click the **Force** check box if you want the unconfigure operation to forcefully remove all of the configuration data. By default, this check box is not selected.

   Note: Select the **Force** check box only if the unconfiguration fails repeatedly. Use this option only as a last resort.

d. Click **Submit** to confirm the operation.

- **Unconfigure a local policy server with a local user registry**
  a. Select the **Clear user registry entries** check box if you want the unconfigure operation to remove all Security Access Manager domain, user, and group information. By default, this check box is not selected.
  b. Select the **Force** check box if you want the unconfigure operation to forcefully remove all of the configuration data. By default, this check box is not selected.

     Note: Select the **Force** check box only if the unconfiguration fails repeatedly. Use this option only as a last resort.

  c. Click **Submit** to confirm the operation.
- **Unconfigure a remote policy server**
  a. Select the **Force** check box if you want the unconfigure operation to forcefully remove all of the configuration data. By default, this check box is not selected.

     Note: Select the **Force** check box only if the unconfiguration fails repeatedly. Use this option only as a last resort.

  b. Click **Submit** to confirm operation.

Managing runtime components with web service

Use the runtime environment resource URI to manage runtime components.

Viewing the runtime environment configuration status with web service

To retrieve the runtime environment status with the RESTful web service, issue an HTTP GET command on the runtime environment resource URI.

URL

```
https://{appliance_hostname}/runtime_components
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Name	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the current configuration status.

Parameter	Description
id	The identifier for the runtime component. The value is always **RTE**.
status	The status of the runtime component. Valid values are **Available**, **Unconfigured**, or **Not available**.
statuscode	A code that represents the status of the runtime component. Valid values are: • **0**: Indicates that the status of the runtime component is **Available**. • **1**: Indicates that the status of the runtime component is **Unconfigured**. • **2**: Indicates that the status of the runtime component is **Not available**.
mode	The configured mode for the runtime component. **Note:** When the **mode** is **Local Standalone** or **Remote Standalone**: • The IBM Security Access Manager policies cannot be distributed between different appliances. • The clustering capability is limited to instances that are located on the same appliance. Valid values are: • **Unconfigured** • **Remote**: Policy server is remote. • **Local Standalone**: Policy server and LDAP are local. • **Remote Standalone**: Policy server is local and LDAP is remote.
modecode	A code that represents the configured mode of the runtime component **Note:** When the **modecode** is **1** or **2**: • The IBM Security Access Manager policies cannot be distributed between different appliances. • The clustering capability is limited to instances that are located on the same appliance. Valid values are: • **-1**: Indicates that the configured mode of the runtime component is **Unconfigured**. • **0**: Indicates that the configured mode of the runtime component is **Remote**. • **1**: Indicates that the configured mode of the runtime component is **Local Standalone**. • **2**: Indicates that the configured mode of the runtime component is **Remote Standalone**.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/runtime_components
```

Response:

```
200 ok

[{
"id":"RTE",
"status":"Available",
"statuscode":"0",
"mode":"Local Standalone",
"modecode":"1"
}]
```

Retrieving a runtime configuration file with web service

To retrieve a runtime configuration file with the RESTful web service, issue an HTTP GET command on the advanced configuration URI.

URL

```
https://{appliance_hostname}/advanced_configuration/{resource_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	Name of the configuration file to retrieve. A valid configuration file name is one of `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.

Response

HTTP response code and JSON code that represents the contents of the file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/advanced_configuration/pd.conf
```

Response:

```
200 ok

{"contents":"#\n# Licensed Materials - Property of IBM\n# 5724-C08
\n# .....remainder of the file contents...."}
```

Exporting a runtime configuration file with web service

To export a runtime configuration file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the advanced configuration URI.

URL

```
https://{appliance_hostname}/advanced_configuration/{resource_id}?export
```

Notes:

- The "`export`" parameter is what differentiates the retrieve operation and the export operation. If the "`export`" parameter is included in the URL, the entire file is exported. If no "`export`" parameter is included in the URL, the content of the file is retrieved.
- The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	Name of the configuration file to export. A valid configuration file name is one of `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.

Response

HTTP response code and file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/advanced_configuration/pd.conf?export
```

Response:

```
200 ok

"#\n# Licensed Materials - Property of IBM\n# 5724-C08
\n# .....remainder of the file contents...."
```

Updating a runtime configuration file with web service

To update a runtime configuration file with the RESTful web service, issue an HTTP PUT command on the advanced configuration URI.

URL

```
https://{appliance_hostname}/advanced_configuration/{resource_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	Name of the configuration file to update. A valid configuration file name is one of `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
file_contents	The new file contents.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/advanced_configuration/{resource_id}

PUT_DATA: {
"file_contents": "file contents to import" }
```

Response:

```
200 ok
```

Reverting a previously updated runtime configuration file with web service

To revert a runtime configuration file with the RESTful web service, issue an HTTP PUT command on the advanced configuration URI.

URL

```
https://{appliance_hostname}/advanced_configuration/{resource_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	Name of the configuration file to update. A valid configuration file name is one of `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
operation	The type of operation to perform. The value of this parameter is always “revert.”

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/advanced_configuration/{resource_id}

PUT_DATA: {
"operation": "revert"
}
```

Response:

```
200 ok
```

Configuring the runtime environment with web service

To configure the runtime environment with the RESTful web service, issue an HTTP POST command on the runtime environment resource URI.

URL

```
https://{appliance_hostname}/runtime_components
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
POST
```

Parameters

Table 4. Configuration parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
ps_mode	The mode for the policy server. Valid values are **local** and **remote**. This field is required.

Table 4. Configuration parameters (continued)

Parameter	Description
user_registry	The type of user registry to be configured against. Valid values are **local**, **ldap**, and **AD**. If the `ps_mode` is **local**, this field must be set to **local**. If the `ps_mode` is **remote**, this field must be set to **ldap** or **AD**. This field is required.
ldap_host	The name of the LDAP or active directory server. This field is required unless `ps_mode` is **local** and `user_registry` is **local**.
ldap_port	The port to be used when the system communicates with the LDAP server. This field is required unless: • `ps_mode` is **local** and `user_registry` is **local**. • `ps_mode` is **remote** and `user_registry` is **AD**.
ldap_dn	The DN that is used when the system contacts the user registry. If the `ps_mode` is **local** and `user_registry` is **ldap**, this field is required. Otherwise this field is ignored.
ldap_pwd	The password for the DN. If the `ps_mode` is **local** and `user_registry` is **ldap**, this field is required. Otherwise this field is ignored.
ldap_ssl_db	The KDB file (no path information is required) that contains the certificate that is used to communicate with the user registry. If no keyfile is provided, the SSL is not used when the system communicates with the user registry. If SSL communication is required to either the LDAP or the AD server, this field is required.
ldap_ssl_label	The label of the SSL certificate that is used when the system communicates with the user registry. This option is only valid if the ldap_ss_db option is provided. If SSL communication is required to either the LDAP or the AD server, this field is optional. Otherwise this field is ignored.
ldap_suffix	The LDAP suffix that is used to hold the Security Access Manager secAuthority data. This is an optional field. If the `ps_mode` is **local** and the `user_registry` is **ldap**, this field is optional. Otherwise this field is ignored.
domain	The IBM Security Access Manager domain name. This field is required unless `ps_mode` is **local** and `user_registry` is **local**.
admin_pwd	The security administrator's password (also known as sec_master). This field is required.

Table 4. Configuration parameters (continued)

Parameter	Description
admin_cert_lifetime	The lifetime in days for the SSL server certificate. If `ps_mode` is **local**, this field is required. Otherwise this field is ignored.
ssl_compliance	Specifies whether SSL is compliant with any additional computer security standard. Valid values are **none**, **fips**, **sp800-131-transition**, **sp800-131-strict**, **suite-b-128**, or **suite-b-192**. If `ps_mode` is **local**, this field is optional. Otherwise, this field is ignored.
ad_ssl	Whether to communicate to the active directory server over SSL. Valid values are **true** and **false**. This field is optional if `ps_mode` is **remote** and `user_registry` is **AD**. Otherwise this field is ignored.
isam_host	The name of the host that hosts the IBM Security Access Manager policy server. If `ps_mode` is **remote**, this field is required. Otherwise this field is ignored.
isam_port	The port over which communication with the IBM Security Access Manager policy server takes place. If `ps_mode` is **remote**, this field is required. Otherwise this field is ignored.
ad_domain	The name of the active directory domain. If `ps_mode` is **remote** and `user_registry` is **AD**, this field is required. Otherwise this field is ignored.
ad_multi_domain	Whether the environment is configured against a multi-domain active directory. Valid values are **true** and **false**. This field is optional if `ps_mode` is **remote** and `user_registry` is **AD**. Otherwise this field is ignored.
ad_dn	The DN within active directory which is used to store the IBM Security Access Manager data. If the `ps_mode` is **remote**, `user_registry` is **AD** and `ad_multi_domain` is **false**, this field is required. Otherwise this field is ignored.
ad_alt_user	Security Access Manager supports the use of an email address (alternate format of the userPrincipalName attribute of the Active Directory user object) as its user ID. This boolean controls whether the alternate format is enabled. Valid values are **true** and **false**. This field is optional if `ps_mode` is **remote** and `user_registry` is **AD**. Otherwise this field is ignored.
ad_global_catalog	The name of the Active Directory global catalog server. If the `ps_mode` is **remote**, `user_registry` is **AD** and `ad_alt_user` is **true**, this field is required. Otherwise this field is ignored.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Remote policy server with remote LDAP user registry

```
POST https://{appliance_hostname}/runtime_components

POST_DATA:
{"ps_mode":"remote",
"user_registry":"ldap",
"isam_host":"isam_host",
"isam_port":"isam_port",
"domain":"isam_domain",
"ldap_host":"ldap_host",
"ldap_port":"ldap_port"}
```

Remote policy server with remote AD user registry

```
POST https://{appliance_hostname}/runtime_components

POST_DATA:
{"ps_mode":"remote",
"user_registry":"AD",
"ldap_host":"ldap_host",
"domain":"isam_domain",
"isam_host":"isam_host",
"isam_port":"isam_port",
"ad_domain":"ad_domain",
"ad_dn":"ad_dn",
"ad_multi_domain":"false",
"ad_alt_user":"true",
"ad_global_catalog":"ad_global_catalog",
"ad_ssl":"true",
"ldap_ssl_db":"ldap_ssl_db",
"ldap_ssl_label":"ldap_ssl_label"}
```

Local policy server with remote LDAP user registry

```
POST https://{appliance_hostname}/runtime_components

POST_DATA:
{"ps_mode":"local",
"user_registry":"ldap",
"ldap_host":"ldap_host",
"ldap_port":"ldap_port",
"domain":"isam_domain",
"ldap_dn":"ldap_dn",
"ldap_pwd":"ldap_pwd",
"ldap_suffix":"ldap_suffix",
"admin_cert_lifetime":"admin_cert_lifetime",
"ssl_compliance":"none",
"admin_pwd":"admin_pwd",
"ldap_ssl_db":"ldap_ssl_db",
"ldap_ssl_label":"ldap_ssl_label"}
```

Local policy server with local user registry

```
POST https://{appliance_hostname}/runtime_components

POST_DATA:
{"ps_mode":"local",
"user_registry":"local",
"admin_cert_lifetime":"admin_cert_lifetime",
"ssl_compliance":"none",
"admin_pwd":"admin_pwd"}
```

Response:

```
200 ok
```

Unconfiguring the runtime environment with web service

To unconfigure the runtime environment component of the appliance with the RESTful web service, issue an HTTP PUT command on the runtime environment resource URI.

URL

```
https://{appliance_hostname}/runtime_components/RTE
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Table 5. Unconfiguration parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
operation	This indicates what the update operation is. The value is always **unconfigure**. This field is required.
ldap_dn	The DN that is used when the system contacts the user registry. This field is required if `ps_mode` is **local** and `user_registry` is **ldap**.
ldap_pwd	The password for the DN. This field is required if `ps_mode` is **local** and `user_registry` is **ldap**.
clean	Whether the unconfigure operation removes all IBM Security Access Manager domain, user, and group information. This field is optional if `ps_mode` is **local**. Otherwise this field is ignored. Valid values are "true" and "false".
force	This option is used to force the unconfiguration if it is failing. For example, this happens when the LDAP server is corrupted. This field is optional. Valid values are "true" and "false".

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Remote policy server

```
PUT https://{appliance_hostname}/runtime_components/RTE

PUT_DATA:
{"operation":"unconfigure",
"force":"true/false"}
```

Local policy server with remote LDAP user registry

```
PUT https://{appliance_hostname}/runtime_components/RTE

PUT_DATA:
{"operation":"unconfigure",
"force":"true/false",
"clean":"true/false",
"ldap_dn":"ldap_dn",
"ldap_pwd":"ldap_pwd"}
```

Local policy server with local user registry

```
PUT https://{appliance_hostname}/runtime_components/RTE

PUT_DATA:
{"operation":"unconfigure",
"force":"true/false",
"clean":"true/false"}
```

Response:

```
200 ok
```

Configuration entry and file management for runtime components with web service

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage configuration entry and configuration file settings for runtime components.

Retrieving a specific configuration entry with web service:

To retrieve a specific configuration entry with the RESTful web service, issue an HTTP GET command on the configuration entry management URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza/
/{stanza_id}/entry_name/{entry_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. A valid configuration file name is one of `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	Matches the stanza in the configuration file. For example, "[server]" without the brackets "[]".
entry_id	The name of the configuration entry. For example, https.

Response

HTTP response code and JSON data that represents an array of entry values.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza/
/{stanza_id}/entry_name/{entry_id}
```

Response:

```
200 ok

{
entryName: [
    "entry_value",
    "entry_value2",
    "entry_value3"
    ]
}
```

Retrieving all configuration entries for a stanza with web service:

To retrieve all configuration parameters for a stanza with the RESTful web service, issue an HTTP GET command on the configuration entry management URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	Matches the stanza in the configuration file. For example, "[server]" without the brackets "[]".

Response

HTTP response code and JSON data that represents an array of entry values.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}
```

Response:

```
200 ok

{entryName: "entry_value",
entryName2: "entry_value2",
entryName3: "entry_value3",
...
}
```

Retrieving a list of stanzas with web service:

To retrieve a list of stanzas with the RESTful web service, issue an HTTP GET command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.

Response

HTTP response code and JSON data that represents the list of stanzas.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza
```

Response:

```
200 ok

[
"stanza_id1",
"stanza_id2",
"stanza_id3",
...
]
```

Deleting a value from a configuration entry with web service:

To delete a configuration entry with the RESTful web service, issue an HTTP DELETE command on the configuration entry management URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza
/{stanza_id}/entry_name/{entry_id}/value/{value_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	Matches the stanza in the configuration file. For example, "[server]" without the brackets "[]".
entry_id	The name of the configuration entry. For example, https.
value_id	The value of the configuration entry to delete.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}/entry_name/{entry_id}/value/{value_id}
```

Response:

```
200 ok
```

Deleting a stanza with web service:

To delete a stanza and all configuration entries that are contained in the stanza with the RESTful web service, issue an HTTP DELETE command on the configuration entry management URI. Only stanzas and all configuration entries within the stanza that are not part of the default configuration file can be deleted.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration/
/stanza/{stanza_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	Matches the stanza in the configuration file. For example, "[server]" without the brackets "[]".

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/runtime/{resource_id}/configuration/
/stanza/{stanza_id}
```

Response:

```
200 ok
```

Adding a configuration stanza name with web service:

To add a configuration stanza name with the RESTful web service, issue an HTTP POST command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	The name of the stanza. For example, "pam-resource:test.html".

Response

HTTP response code and JSON data that represents the new stanza ID.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}
```

Response:

```
200 ok

{"id":"new stanza id"}
```

Adding a configuration entry or entries by stanza with web service:

To add one or more configuration entries by stanza with the RESTful web service, issue an HTTP POST command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}/entry_name
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	The name of the resource stanza entry. For example, "server".
entries	An array of entries key value pairs to be added to the stanza.

Response

HTTP response code and JSON data that represents the entry names.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/runtime/{resource_id}/configuration
/stanza/{stanza_id}/entry_name

POST_DATA: {
  entries:  [
      ["entryName1", "value1"],
      ["entryName2", "value2"]
     ]
  }
```

Response:

```
200 ok

[{"id":"entryName1"},{"id":"entryName2"}]
```

Updating a configuration entry or entries by stanza with web service:

To update one or more configuration entries by stanza with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza
/{stanza_id}/entry_name/{entry_name_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
resource_id	The name of the configuration file to update. For example, `pd.conf`, `ldap.conf`, or `activedir_ldap.conf`.
stanza_id	The name of the resource stanza entry. For example, "server".
entry_name_id	The name of the entry to set.
value	The value to set for the entry.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/runtime/{resource_id}/configuration/stanza
/{stanza_id}/entry_name/{entry_name_id}

PUT_DATA: {
"value":"yes"
}
```

Response:

```
200 ok
```

Web reverse proxy

Use either the local management interface or web service to manage web reverse proxy settings.

There are instance-specific and common settings of the web reverse proxy that you can manage.

The following tasks are instance-specific, which means that the changes you make apply to a particular web reverse proxy instance.

- Instances management and configuration
- Administration pages management
- Routing file management
- Web content protection management
- Tracing management
- Transaction logging management
- Statistics gathering management
- Junction management

The following tasks are common, which means the settings you make apply to all web reverse proxy instances.

- SSO key management
- SSL certificate management
- Query contents management
- URL mapping management
- Junction mapping management
- Client certificate mapping management
- Forms based single sign-on management
- HTTP transformation management

Managing reverse proxy with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.

Management of instances

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to control and configure your web reverse proxy instances.

For detailed information about reverse proxy concepts, see the *IBM Security Access Manager for Web WebSEAL Administration Guide*, SC23-6505-02.

Showing the current state of all instances with local management interface:

To show the current state of all instances with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. You can view the current state and version information of all instances.

Stopping, starting, or restarting an instance with local management interface:

To stop, start or restart an instance with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.

Stop an instance

a. Click **Stop**.
b. A message is displayed indicating that the instance has been stopped successfully.

Start an instance

a. Click **Start**.
b. A message is displayed indicating that the instance has been started successfully.

Restart an instance

a. Click **Restart**.
b. A message is displayed indicating that the instance has been restarted successfully.

Configuring an instance with local management interface:

To configure an instance with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Click **New**.
3. Provide settings for the fields that are displayed on the **Instance**, **IBM Security Access Manager**, **Transport**, and **User Registry** tabs.
   - On the **Instance** tab:

Field	Description
Instance Name	This is the new instance name, which is a unique name that identifies the instance. Multiple instances can be installed on the same computer system. Each instance must have a unique name.
Host Name	The host name that is used by the IBM Security Access Manager policy server to contact the appliance. The address that corresponds to this host name must match a management interface address of the appliance. The addresses that are associated with the application interface of the appliance cannot be used for communication with the IBM Security Access Manager policy server. Valid values include any valid host name or IP address. For example: libra.dallas.ibm.com
Listening Port	This is the listening port through which the instance communicates with the Security Access Manager policy server.
IP Address for the Primary Interface	The IP address for the logical interface.

   - On the **IBM Security Access Manager** tab:

Field	Description
Administrator Name	The Security Access Manager administrator name.
Administrator Password	The Security Access Manager administrator password.

Field	Description
Domain	The Security Access Manager domain.

- On the **Transport** tab:

Field	Description
Enable HTTP	Specifies whether to accept user requests across the HTTP protocol.
HTTP Port	The port to listen for HTTP requests. This field is only valid if the **Enable HTTP** check box is selected.
Enable HTTPS	Specifies whether to accept user requests across the HTTPS protocol.
HTTPS Port	The port to listen for HTTPS requests. This field is only valid if the **Enable HTTPS** check box is selected.

- On the **User Registry** tab:

Field	Description
Enable SSL	Specifies whether to enable SSL communication between the instance and the LDAP server.
Key File Name	The file that contains the LDAP SSL certificate. This field is only valid if the **Enable SSL** check box is selected.
Certificate Label	The LDAP client certificate label. This field is only valid if the **Enable SSL** check box is selected.
Port	The port number through which to communicate with the LDAP server. This field is only valid if the **Enable SSL** check box is selected.

4. Click **Finish**. A message is displayed indicating that the instance has been configured successfully.

Unconfiguring an instance with local management interface:

To unconfigure an instance with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance to unconfigure.
3. Click **Delete**.
4. Enter the administrator name and password.
5. Click **Delete**

   Note: Select the **Force** check box if unconfiguration fails multiple times. Use this option only as a last resort.

Configuration entry and file management

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage configuration entry and configuration file settings.

Managing web reverse proxy configuration entries with local management interface:

To manage the web reverse proxy basic configuration, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Edit**.
4. Make your changes to the settings on the **Server**, **SSL**, **Junction**, **Authentication**, **SSO**, **Session**, **Response**, **Logging**, and **Interfaces** tabs.

Server The Server tab contains entries that are related to the general server configuration.

Field	Description
HTTPS	Select this check box to enable the HTTPS port within Reverse Proxy.
HTTPS Port	The port over which Reverse Proxy listens for HTTPS requests.
HTTP	Select this check box to enable the HTTP port within Reverse Proxy.
HTTP Port	The port over which Reverse Proxy listens for HTTP requests.
Interface Address	The network interface on which the Reverse Proxy server listens for requests.
Persistent Connection Timeout	The maximum number of seconds that a persistent connection with a client can remain inactive before it is closed by the server.
Worker Threads	The number of threads which are allocated to service requests.
Cluster is Master	If the Reverse Proxy clustering function is used, this check box controls whether this Reverse Proxy server acts as the cluster master.
Master Instance Name	The server name for the Reverse Proxy instance which is acting as the master within the cluster. This option is only enabled if the **Cluster is Master** check box is not selected.
Message Locale	The locale in which the Reverse Proxy runs.

SSL The SSL tab contains entries that are related to the general SSL configuration of the server.

Field	Description
SSL Certificate Key File	The key database which is used to store the certificates which are presented by Reverse Proxy to the client.
SSL Server Certificate	The name of the SSL certificate, within the key database, which is presented to the client.
JCT Certificate Key File	The key database which is used to store the certificates which are presented by Reverse Proxy to the junctioned Web servers.

Junction

The Junction tab contains entries that are related to the general junction configuration.

Field	Description
HTTP Timeout	Timeout in seconds for sending to and reading from a TCP junction.

Field	Description
HTTPS Timeout	Timeout in seconds for sending to and reading from an SSL junction.
Ping Interval	The interval in seconds between requests which are sent by Reverse Proxy to junctioned Web servers to determine the state of the junctioned Web server.
Ping Method	The HTTP method that Reverse Proxy uses when it sends health check requests to the junctioned Web server.
Ping URI	The URI that Reverse Proxy uses when it sends health check requests to the junctioned Web server.
Maximum Cached Persistent Connections	The maximum number of connections between Reverse Proxy and a junctioned Web server that will be cached for future use.
Persistent Connection Timeout	The maximum length of time, in seconds, that a cached connection with a junctioned Web server can remain idle before it is closed by Reverse Proxy.
Managed Cookie List	A pattern-matched and comma-separated list of cookie names for those cookies which are stored in the Reverse Proxy cookie jar. Other cookies are passed by Reverse Proxy back to the client.

Authentication

The Authentication tab contains entries that are related to the configuration of the authentication mechanisms which are used by the server.

Basic Authentication

Field	Description
Transport	The transport over which basic authentication is supported.
Realm Name	Realm name for basic authentication.

Forms Authentication

Field	Description
Forms Authentication	The transport over which forms authentication is supported.

Client Certificate Authentication

Field	Description
Accept Client Certificates	Defines the condition under which client certificates are required by Reverse Proxy.
Certificate EAI URI	The resource identifier of the application that is invoked to perform external client certificate authentication.
Certificate Data	The client certificate data that are passed to the EAI application.

EAI Authentication

Field	Description
Transport	The transport over which EAI authentication is supported.
Trigger URL	A URL pattern that is used by Reverse Proxy to determine whether a response is examined for EAI authentication headers.
Authentication Levels	The designated authentication level for each of the configuration authentication mechanisms.

Session

The Session tab contains entries that are related to the general session configuration.

Field	Description
Re-authentication for Inactive	Whether to prompt users to re-authenticate if their entry in the server credential cache has timed out because of inactivity.
Max Cache Entries	The maximum number of concurrent entries in the session cache.
Lifetime Timeout	Maximum lifetime in seconds for an entry in the session cache.
Inactivity Timeout	The maximum time, in seconds, that a session can remain idle before it is removed from the session cache.
TCP Session Cookie Name	The name of the cookie to be used to hold the HTTP session identifier.
SSL Session Cookie Name	The name of the cookie to be used to hold HTTPS session identifier.
Use Same Session	Select the check box to use the same session for both HTTP and HTTPS requests.

Response

The Response tab contains entries that are related to response generation.

Field	Description
Enable HTML Redirect	Select the check box to enable the HTML redirect function.
Enable Local Response Redirect	Select the check box to enable the local response redirect function.
Local Response Redirect URI	When local response redirect is enabled, this field contains the URI to which the client is redirected for Reverse Proxy responses.
Local Response Redirect Macros	The macro information which is included in the local response redirect.

SSO The SSO tab contains entries that are related to the configuration of the different single-sign-on mechanisms that are used by the server.

Failover

Field	Description
Transport	The transport over which failover authentication is supported.

Field	Description
Cookies Lifetime	Maximum lifetime in seconds for failover cookies.
Cookies Key File	The key file which is used to encrypt the failover cookie.

LTPA

Field	Description
Transport	The transport over which LTPA authentication is supported.
Cookie Name	The name of the cookie which is used to transport the LTPA token.
Key File	The key file that is used when accessing LTPA cookies.
Key File Password	The password that is used to access the LTPA key file.

CDSSO

Field	Description
Transport	The transport over which CDSSO authentication is supported.
Transport (generation)	The transport over which the creation of CDSSO tokens is supported.
Peers	The name of the other Reverse Proxy servers that are participating in the CDSSO domain. Along with the name of the keyfile that are used by the Reverse Proxy servers.

ECSSO

Field	Description
Transport	The transport over which e-community SSO authentication is supported.
Name	Name of the e-community.
Is Master Authentication Server	Select the check box if this Reverse Proxy server is the master for the e-community.
Master Authentication Server	The name of the Reverse Proxy server that acts as the master of the e-community. This field is not required if this Reverse Proxy server is designated as the master.
Domain Keys	The name of the other Reverse Proxy servers which are participating in the e-community. Along with the name of the keyfile that is used by the various Reverse Proxy servers.

Logging

The Logging tab contains entries that are related to the logging and auditing configuration.

Field	Description
Enable Agent Logging	Select the check box to enable the agent log.
Enable Referer Logging	Select the check box to enable the referrer log.
Enable Request Logging	Select the check box to enable the request log.

Field	Description
Request Log Format	The format of the entries that are contained within the request log.
Maximum Log Size	The maximum size of the log file before it is rolled over.
Flush Time	The period, in seconds, that Reverse Proxy caches the log entries before the system writes the entries to the log file.
Enable Audit Log	Select the check box to enable the generation of audit events.
Audit Log Type	Select the events to be audited.
Audit Log Size	The maximum size of the audit log file before it is rolled over.
Audit Log Flush	The period, in seconds, that Reverse Proxy caches the audit log entries before the system writes the entries to the log file.

Interfaces

The Interfaces tab contains settings that are related to WebSEAL secondary interfaces.

- To add a new secondary interface, click **New**. Then, define your settings in the pop-up window that contains the following fields:

Field	Description
Application Interface IP Address	The IP address on which the WebSEAL instance listens for requests.
HTTP Port	This field contains the port on which the WebSEAL instance listens for HTTP requests.
HTTPS Port	This field contains the port on which the WebSEAL instance listens for HTTPS requests.
Web HTTP Port	This is the port that the client perceives WebSEAL to be using.
Web HTTP Protocol	This is the protocol that the client perceives WebSEAL to be using.
Certificate Label	The label of the SSL server certificate that is presented to the client by the WebSEAL instance.
Accept Client Certificates	Defines the condition under which client certificates are required by WebSEAL.
Worker Threads	The number of threads that is allocated to service requests.

  Click **Save** to save the settings.
- To delete a secondary interface, select the interface and then click **Delete**.
- To edit a secondary interface, select the interface and click **Edit**. Then, update your settings in the pop-up window that contains the fields that described previously.

5. Click **Save** to apply the changes.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing web reverse proxy configuration files with local management interface:

To manage reverse proxy configurations with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Configuration** > **Edit Configuration File**.
4. Edit the configuration file that is displayed and then click **Save** to save the changes. If you do not want to save the changes, click **Cancel**. If you want to revert to the previous version of the configuration file, click **Revert**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Work with the administration pages root with local management interface

To manage files and directories in the administration pages root with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Management Root**. All current management files and directories are displayed. The default directories include:

   management
   The Web Reverse proxy management pages. For example, `login.html`

   errors The error pages that can be returned by the Web Reverse proxy.

   oauth The HTML files that can be returned by the oauth module.

   junction-root
   The static HTML files that are served by the local junction of the Web Reverse proxy.

   Note: A fixed location is used as the document root. A local junction cannot run any CGI scripts. It can serve only static page content.
4. Work with all the management files and directories.
   - **Create a new file in the administration pages root**
     a. Select the directory in which you want to create the file.
     b. Select **File** > **New** > **File**.
     c. Enter the file name.
     d. Optionally, you can add file contents in the **New File Contents** field.
     e. Click **Save**.

     Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.
   - **Create a new directory in the administration pages root**

a. Select the directory in which to create the directory.
b. Select **File** > **New** > **Directory**.
c. Enter the directory name.
d. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

- **View or update the contents of a file in the administration pages root**
  a. Select the file of interest.
  b. Select **File** > **Open**. You can then view the contents of the file.
  c. Optionally, edit the contents of the file. Then, click **Save**.

  Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.
- **Export a file from the administration pages root**
  a. Select the file of interest.
  b. Select **Manage** > **Export**.

  Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
  c. Confirm the save operation when your browser displays a confirmation window.
- **Rename a file or directory in the administration pages root**
  a. Select the file or directory of interest.
  b. Select **Manage** > **Rename**.
  c. Enter the new name of the file or directory in the **New Resource Name** field.
  d. Click **Save**.

  Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.
- **Delete a file or directory in the administration pages root**
  a. Select the file or directory of interest.
  b. Select **Manage** > **Delete**.
  c. Click **Yes** to confirm the delete operation.

  Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.
- **Import a file to administration pages root**
  a. Select the directory that you want to import the file into.
  b. Select **Manage** > **Import**.
  c. Click **Browse**.
  d. Browse to the file you want to import and then click **Open**.
  e. Click **Import**.
- **Import the contents of a .zip file into the administration pages root**
  a. Select **Manage** > **Import Zip**.
  b. Click **Browse**.

c. Browse to the .zip file you want to import and then click **Open**.
d. Click **Import**.

- **Export the contents of the administration pages root as a .zip file**
  a. Select **Manage** > **Export Zip**.

     Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
  b. Confirm the save operation when your browser displays a confirmation window.

Tracing control

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to control tracing.

Trace data is intended primarily for use by IBM Software Support. Trace data might be requested as part of diagnosing a reported problem. However, experienced product administrators can use trace data to diagnose and correct problems in an IBM Security Access Manager environment. For more information about trace event logging, see *IBM Security Access Manager Troubleshooting Guide*.

Note: Use trace with caution. It is intended as a tool to use under the direction of IBM Software Support. Messages from tracing are sometimes cryptic, are not translated, and can severely degrade system performance.

Listing all trace components and status with local management interface:

To list all trace components, their current levels, and trace file sizes with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Tracing**. All trace components, their current levels, and trace file sizes are then displayed.

Modifying the trace level, flush interval, and rollover size for a component with local management interface:

To modify the trace level, flush interval, and rollover size for a component, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Tracing**. All trace components, their current levels, and trace file sizes are then displayed.
4. Select the component to be modified and then click **Edit**.
5. Modify the trace level, flush interval, and rollover size.
6. Click **Save**.

Managing the trace files for a component with local management interface:

To manage the trace files and rollover files for a component, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Tracing**.
4. Select the component whose trace files and rollover files you want to list. Then, click **Files**. The file name, file size, and last modified time information of all the trace files and rollover files of this component is displayed.

   View or export a trace file or rollover file

   a. Select the file of interest.
   b. Click **View**. The content of the trace files is then displayed. To view a particular number of lines of trace, provide a value in the **Number of lines to view** field and then click **Reload**. Optionally, you can provide a value in the **Starting from line** field to define the start of the lines. If the **Starting from line** field is set, then the **Number of lines to view** field determines how many lines to view forward from the starting line. If the **Starting from line** field is not set, then the **Number of lines to view** field determines how many lines to view from the end of the log file.

      Note: The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
   c. Click **Export** if you want to export the file.

      Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
   d. Confirm the save operation when the browser prompts you to save the file.

   Delete a trace file or rollover file

   a. Select the file of interest.

      Note: Only a file that is not in use can be deleted.
   b. Click **Delete**.
   c. Click **Yes** to confirm the operation.

   Export a trace file or rollover file

   a. Select the file of interest.
   b. Click **Manage** > **Export**.

      Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
   c. Confirm the save operation when the browser prompts you to save the file.

Delete all trace files and rollover files that are not in use

a. Click **Manage** > **Delete All**.

b. Click **Yes** to confirm the operation.

Logging

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage the reverse proxy log files.

Listing the names of all log files and file sizes with local management interface:

To list the names of all log files and file size with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. *Optional:* If instance-specific log files are of interest, select the instance.
3. Select **Manage** > **Logging**. If an instance is selected, details of all common log files and instance-specific log files are displayed. If no instance is selected, only details of the common log files are displayed.

   You can use the filter bar under **Name** to filter entries that meet specific conditions. Click **Clear filter** to return to the full list.

Viewing a snippet of or export a log file with local management interface:

To view a snippet of a log file or export a log file with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. *Optional:* If instance-specific log files are of interest, select the instance.
3. Select **Manage** > **Logging**.
4. Select the log file that you want to view.
5. Click **View**. The content of the log file is displayed. By default, the last 100 lines of a log file is displayed if the file is longer than 100 lines. You can define the number of lines to display by entering the number in the **Number of lines to view** field and then click **Reload**. Optionally, you can provide a value in the **Starting from line** field to define the start of the lines. If the **Starting from line** field is set, then the **Number of lines to view** field determines how many lines to view forward from the starting line. If the **Starting from line** field is not set, then the **Number of lines to view** field determines how many lines to view from the end of the log file.

   Note: The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
6. Click **Export** to download the log file.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.

   You can also export a file by selecting it and then clicking **Manage** > **Export**.

Clearing a log file with local management interface:

To clear a log file and turn its size to 0 with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. *Optional:* If instance-specific log files are of interest, select the instance.
3. Select **Manage** > **Logging**.
4. Select the log file that you want to clear.
5. Click **Clear**.
6. On the Confirm Action confirmation page, click **Yes**.

Statistics control

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage statistics.

Retrieving all statistics components and their details with local management interface:

To retrieve all statistics components details with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Statistics**. All statistics components and their status, interval, count, total accumulated statistics log file size, flush interval, and rollover size are displayed.

Modifying the statistics settings for a component with local management interface:

To modify the statistics settings for a particular component with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Statistics**.
4. Select the statistics component that you want to modify.
5. Click **Edit**.
6. Select the check box beside **Enabled** if it is not already checked.
7. Modify the **Interval**, **Count**, **Flush Interval**, and **Rollover Size** fields as needed.
8. Click **Save** to save your changes.

Managing statistics log files with local management interface:

To manage statistics log files with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Statistics**.
4. Select the statistics component of interest.
5. Click **Files**. The file name, file size, and last modified time information of all statistics log files is displayed.
   - **View a statistics log file or a snippet of a statistics log file**
     a. Select the statistics log file that you want to view and then click **View**. The contents of the statistics log file are displayed.
     b. You can enter a value into the **Number of lines to view** field and then click **Reload** to get a customized snippet view of the log file. Optionally, you can provide a value in the **Starting from line** field to define the start of the lines. If the **Starting from line** field is set, then the **Number of lines to view** field determines how many lines to view forward from the starting line. If the **Starting from line** field is not set, then the **Number of lines to view** field determines how many lines to view from the end of the log file.

        Note: The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
   - **Export a statistics log file**
     a. Select the statistics log file that you want to export.
     b. Click **Manage** > **Export**.

        Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
     c. Confirm the save operation in the browser window displayed.
   - **Delete a statistics log file**
     a. Select the statistics log file that you want to delete and then click **Delete**.

        Note: Only log files that are not in use can be deleted. To disable a log file, you can select the log file, click **Edit**, clear the **Enabled** check box, and then click **Save**.
     b. Click **Yes** to confirm the operation.
   - **Delete all unused statistics log files**
     a. Click **Manage** > **Delete All**.
     b. Click **Yes** to confirm the operation.

Updating a routing control file with local management interface

To update a routing control file with the local management interface, use the Reverse Proxy Instances management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Configuration** > **Edit Routing File**. The routing file contents are displayed.
4. Modify the routing file.
5. Click **Save** to save the changes. Or click **Close** if you do not want to save the changes.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing transaction logging components and data files with local management interface

To manage transaction logging components and data files with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the instance of interest.
3. Select **Manage** > **Troubleshooting** > **Transaction Logging**. All transaction logging components and their status, total file size, and rollover size are displayed.
   - **Enable or disable a transaction logging component**
     a. Select the transaction logging component of interest.
     b. Click **Edit**.
     c. Select or clear the **Enabled** check box to enable or disable the transaction logging component.
     d. Optionally, define the rollover size by providing a value in the **Rollover Size** field. If no value is provided, the default rollover size is used.
     e. Click **Save** to save your changes.
   - **Rollover the data file of a transaction logging component**
     a. Select the transaction logging component of interest.
     b. Click **Manage** > **Rollover**.
     c. Click **Yes** to confirm the operation.
   - **Manage transaction logging data files**
     a. Select the transaction logging component of interest.
     b. Click **Files**.
        – **Export a transaction logging data file**
          1) Select the transaction logging data file of interest.
          2) Click **Manage** > **Export**.

             Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
          3) Confirm whether to open or save the exported file in the browser window.
        – **Delete a transaction logging data file**

Note: Only transaction logging data files that are not in use can be deleted.

1) Select the transaction logging data file of interest.
2) Click **Delete**.
3) Click **Yes** to confirm the operation.

– **Delete all unused transaction logging data files**

1) Click **Manage** > **Delete All**.
2) Click **Yes** to confirm the operation.

Managing standard and virtual junctions with local management interface

To manage standard and virtual junctions with the local management interface, use the Junction Management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the reverse proxy to manage junctions for.
3. Select **Manage** > **Junction Management**.
4. Perform junction related tasks as needed.
   - **Create a standard junction**
     a. Click **New** > **Standard Junction**.
     b. On the Junction tab page:
        1) Enter the junction point name.

           Note: Names for standard junctions must start with a forward slash (/) character.
        2) Select the **Create Transparant Path Junction** check box if the junction name must match the name of a subdirectory under the root of the back-end server document space.
        3) Select the **Stateful Junction** check box if you want the junction to be stateful.
        4) Select a junction type from the listed options.
     c. On the Servers tab page:
        1) Click **New** to add a target back-end server.

           Note: At lease one target back-end server must be added to create a junction. The options available when you add a server vary depending on the junction type selected.
        2) Complete the fields displayed.
        3) Click **Save**.
     d. On the Basic Authentication tab page:
        1) Select the **Enable Basic Authentication** check box if BA header information is to be used for authentication with the back-end server.
        2) Enter the WebSEAL user name in the **Username** field.
        3) Enter the WebSEAL password in the **Password** filed.
        4) Select the **Enable mutual authentication to junctioned WebSEAL servers** check box if mutual authentication is to be used between a frontend WebSEAL server and a back-end WebSEAL server.

5) Select the key file from the list to use for mutual authentication.
6) Select the key label from the list to use for mutual authentication.

e. On the Identity tab page:
1) Define how WebSEAL server passes client identity information in BA headers to the back-end server by selecting appropriate actions from the list under **HTTP Basic Authentication Header**.
2) If **GSO** is selected in the previous step, enter the GSO resource or resource group name in the **GSO Resource or Group** field. If a value other than **GSO** is selected in the previous step, skip this step.
3) Select what HTTP header identity information is passed to the back-end server in the **HTTP Header Identity Information** field.
4) Select encoding from the list under **HTTP Header Encoding**.
5) Select an option from the list under **Junction Cookie Javascript Block**.
6) Select the check box on the right as necessary.

f. On the SSO and LTPA tab page:
1) Select the **Enable LTPA cookie Support** check box if the junctions are to support LTPA cookies.
2) If LTPA version 2 cookies (LtpaToken2) are used, select the **Use Version 2 Cookies** check box.
3) Select the LTPA keyfile from the list under **LTPA Keyfile**.
4) Enter the keyfile password in the **LTPA Keyfile Password** field.

g. On the General tab page:
1) Specify the name of the form based single sign-on configuration file in the **FSSO Configuration File** field.
2) Define the hard limit for consumption of worker threads in the **Percentage Value for Hard Limit of Worker Threads** field.
3) Define the soft limit for consumption of worker threads in the **Percentage Value for Soft Limit of Worker Threads** field.
4) If you want denied requests and failure reason information from authorization rules to be sent in the Boolean Rule header, select the **Include authorization rules decision information** check box.

h. Click **Save**.

- **Create a virtual junction**

a. Click **New** > **Virtual Junction**.
b. On the Junction tab page:
1) Enter the junction label in the **Junction Label** field.
2) Select the **Stateful Junction** check box if you want the junction to be stateful.
3) Select a junction type from the listed options on the right.

c. On the Servers tab page:
1) Click **New** to add a target back-end server.

Note: At lease one target back-end server must be added to create a junction.

2) Complete the fields displayed.
3) Click **Save**.

d. On the Basic Authentication tab page:

1) Select the **Enable Basic Authentication** check box if BA header information is to be used for authentication with the back-end server.
2) Enter the WebSEAL user name in the **Username** field.
3) Enter the WebSEAL password in the **Password** filed.
4) Select the **Enable mutual authentication to junctioned WebSEAL servers** check box if mutual authentication is to be used between a frontend WebSEAL server and a back-end WebSEAL server.
5) Select the key file from the list to use for mutual authentication.
6) Select the key label from the list to use for mutual authentication.

e. On the Identity tab page:
1) Define how WebSEAL server passes client identity information in BA headers to the back-end server by selecting appropriate actions from the list under **HTTP Basic Authentication Header**.
2) If **GSO** is selected in the previous step, enter the GSO resource or resource group name in the **GSO Resource or Group** field. If a value other than **GSO** is selected in the previous step, skip this step.
3) Select what HTTP header identity information is passed to the back-end server in the **HTTP Header Identity Information** field.
4) Select encoding from the list under **HTTP Header Encoding**.
5) Select the check box on the right as necessary.

f. On the SSO and LTPA tab page:
1) Select the **Enable LTPA cookie Support** check box if the junctions are to support LTPA cookies.
2) If LTPA version 2 cookies (LtpaToken2) are used, select the **Use Version 2 Cookies** check box.
3) Select the LTPA keyfile from the list under **LTPA Keyfile**.
4) Enter the keyfile password in the **LTPA Keyfile Password** field.

g. On the General tab page:
1) Specify the name of the form based single sign-on configuration file in the **FSSO Configuration File** field.
2) Define the hard limit for consumption of worker threads in the **Percentage Value for Hard Limit of Worker Threads** field.
3) Define the soft limit for consumption of worker threads in the **Percentage Value for Soft Limit of Worker Threads** field.
4) If you want denied requests and failure reason information from authorization rules to be sent in the Boolean Rule header, select the **Include authorization rules decision information** checkbox.

h. Click **Save**.

- **Edit a standard or virtual junction**
  a. Select the junction to edit from the list.
  b. Click **Edit**.
  c. Modify the settings as needed.
  d. Click **Save**.
- **Delete a standard or virtual junction**
  a. Select the junction to delete from the list.
  b. Click **Delete**.
  c. In the confirmation window that pops up, click **Yes**.

Note: Some junction management tasks can be performed only with the web service, but not the local management interface. For example, functions achieved by using the following web service commands cannot be achieved by using the local management interface:

- jmt load
- jmt clear
- offline
- online
- throttle

Configuring web application firewall with local management interface

To configure web application firewall with the local management interface, use the Reverse Proxy management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Manage** > **Reverse Proxy**.
2. Select the Reverse Proxy instance to configure web application firewall for.
3. Click **Manage** > **Configuration** > **Web Content Protection**.
4. On the Operating Configuration tab, you can configure general Web Content Protection settings.
   a. Select the **Enable Web Content Protection** check box to enable the web application firewall.
   b. Select the **Use Proxy HTTP Header** check box as needed. This is used to control whether the audit log contains the IP address of the client as obtained from the network connection, or the IP address that is obtained from the x-forwarded-for HTTP header. This setting is useful when a network terminating firewall sits between the reverse proxy and the client.
   c. Provide a value in bytes for the **Maximum Memory Size** field. This defines the maximum memory that can be used by the PAM engine.

      Note: PAM has a pre-defined minimum memory size. If the configured value is set to less than the minimum, the allocated memory is automatically increased to this minimum size.
   d. Under **Resource Actions**:

      Note: Use this table to customize the actions that are taken when issues are encountered for a particular resource. This is a pattern-matched list that is searched in order. The resource name can contain the "*" and "?" pattern-matching characters. If no matching resource is found, the default actions, as recommended by the x-force team, are taken.

      - To add a resource:
        1) Click **New**.
        2) On the Add Custom Resource page, provide the resource name. All issues available to the resource are pre-populated.

           Note: Resource names can contain the "*" and "?" pattern-matching characters. For example, *.html.
        3) Select an issue that you want to modify and then click **Edit**.
        4) On the Edit Custom Resource Issue page, select the action to take against this issue in the **Response** field.

5) *Optional:* If **Quarantine** is selected as the event response in the previous step, specify the quarantine time in the **Quarantine Period** field.
6) Click **Save** on the Edit Custom Resource Issue page.
7) Click **Save** on the Add Custom Resource page.

- To edit a resource:
  1) Select the resource name to edit.
  2) Click **Edit**.
  3) On the Edit Custom Resource page, select the issue that you want to modify and then click **Edit**.
  4) On the Edit Custom Resource Issue page, modify the event response and quarantine time as needed.
  5) Click **Save** on the Edit Custom Resource Issue page.
  6) Click **Save** on the Edit Custom Resource page.
- To delete a resource:
  1) Select the resource name to delete.
  2) Click **Delete**.

     Note: There is no confirmation window for this delete operation. Make sure that the selected resource is the one you want to delete before you click **Delete**.

e. Under **Registered Resources**:

Note: The registered resources are used to designate the requests that are passed to the inspection engine. When a request is received by the Web reverse proxy, the entries in the list is sequentially searched until a match is found. The action that is assigned to the matching resource controls whether the inspection is enabled or disabled. The resources can contain wildcard characters for pattern matching.

- To add a registered resource:
  1) Click **New**.
  2) On the Add Protected Resources page that pops up, provide the **Resource Name**. For example, index.html, *.html or *.gif.
  3) Select **Enabled** or **Disabled** as needed.
  4) Click **Save**.
- To edit a registered resource:
  1) Select the resource to edit from the list.
  2) Click **Edit**.
  3) On the Edit Protected Resources page that pops up, modify the resource name and whether it is enabled as needed.
  4) Click **Save**.
- To delete a registered resource
  1) Select the resource to delete from the list.
  2) Click **Delete**.

     Note: There is no confirmation window for this delete operation. Make sure that the selected resource is the one you want to delete before you click **Delete**.

f. Under **Injection Tuning Parameters**, modify the listed parameters by double-clicking a value in the **Units** column and editing inline as needed. To see a description of each parameter, hover your mouse cursor on that parameter and a pop-up message that contains the description is displayed.

5. On the Issues tab, you can enable or disable certain issues.

   Note: The list of issues control the events that are monitored by the inspection engine. If an issue is disabled, the inspection engine no longer checks for this issue.

   - Approach 1:
     a. Select the event to edit.
     b. Click **Edit**.
     c. On the Edit Issue page, select **Enabled** or **Disabled** as needed.
     d. Click **Save**.
   - Approach 2:
     – Select or clear the **Enabled** check box to enable or disable a particular issue.
   - Approach 3:
     – Click **Trust X-Force** to automatically disable all issues for which there is not a default response.

6. On the Audit tab, you can configure logging and auditing settings.
   a. Under **Log detailed audit events**, select the check box if you want to enable logging for detailed audit events.
   b. Under **Log Audit Events**, select one of the options to indicate where the audit events are sent.
   c. Under **Log Audit Config**, define the following parameters based on the selections made in the previous step.
      - If **Log to File** is selected:

Parameter	Description
File Name	The entry specifies the name of the log file.
Rollover Size	The maximum size to which a log file can grow before it is rolled over. The default value is 2000000 bytes.
Buffer Size	The maximum size of the message that is used when smaller events are combined.
Queue Size	There is a delay between events being placed on the queue and the file log agent removing them. This parameter specifies the maximum size to which the queue is allowed to grow.
High Water Mark	Processing of the event queue is scheduled regularly at the configured flush interval. It also is triggered asynchronously by the queue size reaching a high water mark on the event queue. The default value is two-thirds of the maximum configured queue size. If the maximum queue size is zero, the high water mark is set to a default of 100. If the event queue high water mark is set to 1, every event queued is relayed to the log agent as soon as possible.
Flush Interval	This entry controls the frequency with which the server asynchronously forces a flush of the file stream to disk. The value defined for this parameter is 0, < 0, or the flush interval in seconds.

      - If **Log to Remote Authorization Server** is selected:

Parameter	Description
Compress	To reduce network traffic, use this parameter to compress buffers before transmission and expand on reception. The default value is `no`.
Buffer Size	To reduce network traffic, events are buffered into blocks of the nominated size before they are relayed to the remote server. This parameter specifies the maximum message size that the local program attempts to construct by combining smaller events into a large buffer. The default value is 1024 bytes.
Flush Interval	This parameter limits the time that a process waits to fill a consolidation buffer. The default value is 20 seconds. A flush interval of 0 is not allowed. Specifying a value of 0 results in the buffer being flushed every 600 seconds.
Queue Size	There is a delay between events being placed on the queue and the file log agent removing them. This parameter specifies the maximum size to which the queue is allowed to grow.
High Water Mark	Processing of the event queue is scheduled regularly at the configured flush interval. It also is triggered asynchronously by the queue size reaching a high water mark on the event queue. The default value is two-thirds of the maximum configured queue size. If the maximum queue size is zero, the high water mark is set to a default of 100. If the event queue high water mark is set to 1, every event queued is relayed to the log agent as soon as possible.
Error Retry Timeout	If a send operation to a remote service fails, the system tries again. Before the system tries again, it waits for the error retry timeout in seconds. The default value is 2 seconds.
Logging Port	Configure the port parameter to specify the port that the remote authorization server listens on for remote logging requests. The default value is port 7136.
Rebind Retry	If the remote authorization server is unavailable, the log agent attempts to rebind to this server at this frequency in number of seconds. The default rebind retry timeout value is 300 seconds.
Hostname	The remote logging services are offered by the authorization service. The server parameter nominates the hosts to which the authorization server process is bound for event recording.
DN	To establish mutual authentication of the remote server, a distinguished name (DN) must be configured. A distinguished name must be specified as a string that is enclosed by double quotation marks.

- If **Log to Remote Syslog Server** is selected:

Parameter	Description
Remote Syslog Server	The host to which the syslog server process is bound for event recording.
Port	The port on which the remote syslog server listens for remote logging requests.
Application ID	The name of the application, as it appears in the messages that are sent to the remote syslog server.
Error Retry Timeout	If a send operation to a remote service fails, the system tries again. Before the system tries again, it waits for the error retry timeout in seconds. The default value is 2 seconds.

Parameter	Description
Flush Interval	This parameter limits the time that a process waits to fill a consolidation buffer. The default value is 20 seconds. A flush interval of 0 is not allowed. Specifying a value of 0 results in the buffer being flushed every 600 seconds.
High Water Mark	Processing of the event queue is scheduled regularly at the configured flush interval. It also is triggered asynchronously by the queue size reaching a high water mark on the event queue. The default value is two-thirds of the maximum configured queue size. If the maximum queue size is zero, the high water mark is set to a default of 100. If the event queue high water mark is set to 1, every event queued is relayed to the log agent as soon as possible.
Queue Size	There is a delay between events being placed on the queue and the file log agent removing them. This parameter specifies the maximum size to which the queue is allowed to grow.
Rebind Retry	If the remote system log server is unavailable, the log agent attempts to rebind to this server at this frequency in number of seconds. The default rebind retry timeout value is 300 seconds.
Maximum Event Length	The maximum length of an event to be transmitted to the remote syslog server. If the event text is longer than the configured length, it is truncated to the maximum event length. If the maximum event length is zero, the event text is never truncated. If transmitting the event to the remote syslog server in clear text, set the maximum event length to less than the maximum transmission unit (MTU) for the network path to the server. This avoids fragmentation of the event.
Enable SSL Communication	Whether SSL is be used for communication.
SSL Keyfile	The name of the GSKit key database file that contains the CA certificate. It is used when the system establishes a secure connection with the remote syslog server over TLS. If the **Enable SSL Communication** check box is selected, this field is required.
SSL Certificate Label	The name of the certificate to be presented to the remote syslog server, upon request, when the system establishes a secure connection. If no value is set for this field, the default certificate from the key database is used.

7. On the Advanced Configuration tab, you can configure coalescer, inspection engine, issues, and custom actions.
   a. Under **Coalescer Configuration**:

      Note: The coalescer is used to correlate audit events. The administrator can use these configuration settings to fine-tune the processing of the coalescer and thus reduce the number of messages that are sent to the audit log.

      - To add a coalescer parameter:
        1) Click **New**.
        2) On the Add Coalescer Parameter page that pops up, provide the parameter name and value.
        3) Click **Save**.
      - To edit a coalescer parameter:
        1) Select the parameter to edit from the list.
        2) Click **Edit**.

3) On the Edit Coalescer Parameter page that pops up, modify the parameter name and value as needed.
4) Click **Save**.

- To delete a coalescer parameter:
  1) Select the parameter to delete from the list.
  2) Click **Delete**.

     Note: There is no confirmation window for this delete operation. Make sure that the selected parameter is the one you want to delete before you click **Delete**.

b. Under **Inspection Engine Configuration**:

- To add a inspection engine configuration parameter:
  1) Click **New**.
  2) On the Add Inspection Parameter page that pops up, provide the parameter name and value.
  3) Click **Save**.
- To edit a inspection engine configuration parameter:
  1) Select the parameter to edit from the list.
  2) Click **Edit**.
  3) On the Edit Inspection Parameter page that pops up, modify the parameter name and value as needed.
  4) Click **Save**.
- To delete a inspection engine configuration parameter:
  1) Select the parameter to delete from the list.
  2) Click **Delete**.

     Note: There is no confirmation window for this delete operation. Make sure that the selected resource is the one you want to delete before you click **Delete**.

8. Click **Save**.

Managing reverse proxy with web service

Use the Reverse Proxy resource URI to manage reverse proxies.

Management of instances

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to control and configure your web reverse proxy instances.

Retrieving the current state of all instances with web service:

To retrieve the current state of all instances with the RESTful web service, issue an HTTP GET command on the Reverse Proxy resource URI.

URL

```
https://{appliance_hostname}/reverseproxy
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Name	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the status of all instances.

Parameter	Description
id	An identifier for this instance.
instance_name	The name of this instance.
enabled	Whether this instance is enabled. Valid values are “yes” and “partial”.
started	Whether this instance is started. Valid values are “yes” and “no”.
version	The current version of the instance .conf file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970)
restart	Indicates whether this instance needs to be restarted to allow configuration changes to go into effect.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy
```

Response:

```
200 ok


[{
"id": "instance_1",
"instance_name":"instance_1",
"enabled": "yes",
"started":"yes",
"version":"version_id",
"restart":"true/false"
},
{
"id": "instance_2",
"instance_name":"instance_2",
"enabled": "partial",
"started":"no",
"version":"version_id"
"restart":"true/false"
},
{
....
}]
```

Stopping, starting, or restarting an instance with web service:

To stop, start, or restart a selected instance with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the instance to act on.
operation	A flag that indicates whether the operation required is to stop, start, or restart the instance. Valid values are "stop", "start", and "restart".

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{id}

PUT_DATA: {
"operation":"stop/start/restart"
}
```

Response:

```
200 ok
```

Configuring an instance with web service:

To configure an instance with the RESTful web service, issue an HTTP POST command on the Reverse Proxy resource URI.

URL

```
https://{appliance_hostname}/reverseproxy
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
inst_name	This is the new instance name, which is a unique name that identifies the instance. Multiple instances can be installed on that same computer system. Each instance must have a unique name.
host	The host name that is used by the IBM Security Access Manager policy server to contact the appliance. The address that corresponds to this host name must match a management interface address of the appliance. The addresses that are associated with the application interface of the appliance cannot be used for communication with the IBM Security Access Manager policy server. Valid values include any valid host name or IP address. For example: libra.dallas.ibm.com This parameter is required.
listening_port	This is the listening port through which the instance communicates with the Security Access Manager policy server.
domain	The Security Access Manager domain.
admin_id	The Security Access Manager administrator name.
admin_pwd	The Security Access Manager administrator password.
ssl_yn	Specifies whether to enable SSL communication between the instance and the LDAP server. The value can be set to “yes” or “no”.
key_file	The file that contains the LDAP SSL certificate. This parameter is only valid if **ssl_yn** is set to “yes”.
cert_label	The LDAP client certificate label. This parameter is only valid if **ssl_yn** is set to “yes”.
ssl_port	The port number through which to communicate with the LDAP server. This parameter is only valid if **ssl_yn** is set to “yes”.
http_yn	Specifies whether to accept user requests across the HTTP protocol. The value can be set to “yes” or “no”.
http_port	The port to listen for HTTP requests. This parameter is only valid if **http_yn** is set to “yes”.
https_yn	Specifies whether to accept user requests across the HTTPS protocol. The value can be set to “yes” or “no”.
https_port	The port to listen for HTTPS requests. This parameter is only valid if **https_yn** is set to “yes”.
nw_interface_yn	Specifies whether to use a logical network interface for the instance. The value can be set to “yes” or “no”.
ip_address	The IP address for the logical interface. This parameter is only valid if **nw_interface_yn** is set to “yes”.

Response

HTTP response code and ID of the new instance name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy

POST_DATA: {
"inst_name": "instance_name",
"host":"192.168.152.130",
"listening_port":"7234",
"domain":"Default",
"admin_id":"sec_master",
"admin_pwd":"passw0rd",
"ssl_yn":"yes",
"key_file":"keyfile",
"cert_label":"label",
"ssl_port":"4043",
"http_yn":"yes",
"http_port":"80",
"https_yn":"yes",
"https_port":"443",
"nw_interface_yn":"yes",
"ip_address":"192.168.152.131"
}
```

Response:

```
200 ok

{"id":"new instance name"}
```

Unconfiguring an instance with web service:

To unconfigure an instance with the RESTful web service, issue an HTTP POST command on the Reverse Proxy resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	This is the instance name to unconfigure, which is a unique name that identifies the instance.
admin_id	The Security Access Manager administrator name.
admin_pwd	The Security Access Manager administrator password.
operation	A flag that is used to indicate the operation to perform. This value is set to "unconfigure" for the unconfigure operation.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{id}

PUT_DATA: {
"admin_id":"sec_master",
"admin_pwd":"passw0rd",
"operation":"unconfigure"
}
```

Response:

```
200 ok
```

Configuration entry and file management

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage configuration entry and configuration file settings.

Retrieving a specific configuration entry with web service:

To retrieve a specific configuration entry with the RESTful web service, issue an HTTP GET command on the configuration entry management URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/stanza/
/{stanza_id}/entry_name/{entry_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Description	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.
stanza_id	Matches the stanza in the configuration file. For example, "[server]" without the brackets "[]".
entry_id	The name of the configuration entry. For example, https.

Response

HTTP response code and JSON data that represents an array of entry values.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{reverseproxy_id}
/configuration/stanza//{stanza_id}/entry_name/{entry_id}
```

Response:

```
200 ok

{
entryName: [
    "entry_value",
    "entry_value2",
    "entry_value3"
    ]
}
```

Retrieving all configuration entries for a stanza with web service:

To retrieve all configuration parameters for a stanza with the RESTful web service, issue an HTTP GET command on the configuration entry management URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.
stanza_id	Matches the stanza in the configuration file. For example, "[server]" without the brackets "[]".

Response

HTTP response code and JSON data that represents an array of entry values.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}
```

Response:

```
200 ok

{entryName: "entry_value",
entryName2: "entry_value2",
entryName3: "entry_value3",
...
}
```

Retrieving a list of stanzas with web service:

To retrieve a list of stanzas with the RESTful web service, issue an HTTP GET command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/stanza
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.

Response

HTTP response code and JSON data that represents the list of stanzas.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/stanza
```

Response:

```
200 ok

[
"stanza_id1",
"stanza_id2",
"stanza_id3",
...
]
```

Adding a configuration stanza name with web service:

To add a configuration stanza name with the RESTful web service, issue an HTTP POST command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.
stanza_id	The name of the stanza. For example, "pam-resource:test.html".

Response

HTTP response code and JSON data that represents the new stanza ID.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}
```

Response:

```
200 ok

{"id":"new stanza id"}
```

Adding a configuration entry or entries by stanza with web service:

To add one or more configuration entries by stanza with the RESTful web service, issue an HTTP POST command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}/entry_name
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.
stanza_id	The name of the resource stanza entry. For example, "server".
entries	An array of entries key value pairs to be added to the stanza.

Response

HTTP response code and JSON data that represents the entry names.

Note: For descriptions of possible error responses that can be returned from a web service call, see"Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}/entry_name

POST_DATA: {
  entries:  [
      ["entryName1", "value1"],
      ["entryName2", "value2"]
    ]
  }
```

Response:

```
200 ok

[{"id":"entryName1"},{"id":"entryName2"}]
```

Updating a configuration entry or entries by stanza with web service:

To update one or more configuration entries by stanza with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/stanza
/{stanza_id}/entry_name/{entry_name_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.
stanza_id	The name of the resource stanza entry. For example, “server”.
entry_name_id	The name of the entry to set.
value	The value to set for the entry.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/stanza
/{stanza_id}/entry_name/{entry_name_id}

PUT_DATA: {
"value":"yes"
}
```

Response:

```
200 ok
```

Deleting a value from a configuration entry with web service:

To delete a configuration entry with the RESTful web service, issue an HTTP DELETE command on the configuration entry management URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/stanza
/{stanza_id}/entry_name/{entry_id}/value/{value_id}
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in “Configuration changes commit process” on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.

Parameter	Description
stanza_id	Matches the stanza in the configuration file. For example, “[server]” without the brackets “[]”.
entry_id	The name of the configuration entry. For example, https.
value_id	The value of the configuration entry. This field is optional and if it is not specified, all configuration entries with the matching stanza_id and entry_id are deleted.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration
/stanza/{stanza_id}/entry_name/{entry_id}/value/{value_id}
```

Response:

```
200 ok
```

Deleting a stanza with web service:

To delete a stanza and all configuration entries that are contained in the stanza with the RESTful web service, issue an HTTP DELETE command on the configuration entry management URI. Only stanzas and all configuration entries within the stanza that are not part of the default configuration file can be deleted.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/
/stanza/{stanza_id}
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in “Configuration changes commit process” on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The name of the reverse proxy instance to update.
stanza_id	Matches the stanza in the configuration file. For example, “[server]” without the brackets “[]”.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/configuration/
/stanza/{stanza_id}
```

Response:

```
200 ok
```

Administration pages management

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage administration pages.

Managing the web reverse proxy security policy:

The management of web reverse proxy security policy includes managing users, web reverse proxy instances (for example, junctions), and security policies (for example, ACLs and POPs).

There are four ways in which the web reverse proxy security policy can be managed:

- Install IBM Security Access Manager runtime environment on an external machine and configure it against the policy server that is in use by the appliance. Then, manage the web reverse proxy security policy with the **pdadmin** command.
- Install the IBM Security Access Manager - Web Portal Manager on an external machine. Configure it against the policy server that is in use by the appliance. Then, manage the web reverse proxy security policy with the Web Portal Manager GUI console.
- Use the command-line interface that is provided by the appliance. This method does not require installation of software on an external machine.
- Use a web service to issue **pdadmin** commands. For more information, see "Running pdadmin commands with web service" on page 89.

To manage the web reverse proxy security policy with the command-line interface provided by the appliance, follow these steps:

1. Access the command-line interface of the appliance by using either the console or a remote ssh session. Use the appliance administration user (`admin`) in the authentication process.
2. Enter `wga` to work with web gateway settings.
3. Enter `admin` to start an administration session, which invokes the **pdadmin** command shell. See the *IBM Security Access Manager Administration Guide* for detailed references on the usage of the **pdadmin** command.

   Note: You can start the administration session only if the web reverse proxy runtime environment is configured. For more information about configuring web reverse proxy runtime environment, see "Configure the runtime

environment with local management interface" on page 31 or "Configuring the runtime environment with web service" on page 39.

4. Log in to your policy server with the **sec_master** user or a user that has authority to administer the policy server.
5. Use the **pdadmin** command shell to change your security policy settings as needed.
6. Enter `quit` to exit the administration session.

Running pdadmin commands with web service:

To run one or multiple pdadmin commands with the RESTful web service, issue an HTTP POST command on the pdadmin resource URI.

URL

```
https://{appliance_hostname}/pdadmin
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

in the format of

```
POST_DATA:
{
"admin_id":"The administrator id",
"admin_pwd":"The administrator password",
"commands":
[
"The first command to run",
"The second command to run",
...
]
}
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
admin_id	The IBM Security Access Manager administrator name. This parameter is required.
admin_pwd	The IBM Security Access Manager administrator password. This parameter is required.
commands	The pdadmin commands to run. This parameter is required.

Response

HTTP response code and JSON data that represents the output from the pdadmin commands.

If you run multiple pdadmin commands, a failure response is only returned if the last command fails. If a command other than the last fails, the response is successful. View the result to ensure that the previous commands are all successful. Also, if a single command fails, it does not halt the operation of subsequent commands.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/pdadmin

POST_DATA:
{
"admin_id":"sec_master",
"admin_pwd":"password",
"commands":
[
"server list"
]
}
```

Response:

```
200 ok

{"result":"cmd> server list\n ivmgrd-master\n test-webseald-appliance"}
```

Retrieving the current administration pages root contents with web service:

To retrieve the current administration pages root contents with the RESTful web service, issue an HTTP GET command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
instance_id	ID of the relevant instance.

Response

JSON data that represents the current administration pages root contents.

```
{
"id": "0",
"name":"management",
"type":"Directory",
"version":"version_id"
},
{
"id": "1",
"name":"errors",
"type":"Directory",
"version":"version_id"
}
```

```
{
"id": "2",
"name":"oauth",
"type":"Directory",
"version":"version_id"
},
{
"id": "3",
"name":"junction_root",
"type":"Directory",
"version":"version_id"
}
```

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Resource index of the administration pages root contents.
name	Name of the administration pages root contents.
type	Resource type. Valid values are “file” and “directory”.
version	The current version of the file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

Retrieve the contents of the administration pages root non-recursively

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/management_root
```

Retrieve the contents of a sub directory

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/
management_root/management
```

Retrieve the contents of the administration pages root or a sub directory recursively

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root?recursive=yes

GET https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root/management?recursive=yes
```

Response:

Retrieve the contents of the administration pages root non-recursively

```
200 ok

{
"id": "0",
"name":"management",
"type":"Directory",
"version":"version_id"
},
{
"id": "1",
"name":"errors",
```

```
"type":"Directory",
"version":"version_id"
}
{
"id": "2",
"name":"oauth",
"type":"Directory",
"version":"version_id"
},
{
"id": "3",
"name":"junction_root",
"type":"Directory",
"version":"version_id"
}
```

Retrieve the contents of the administration pages root or a sub directory recursively

```
200 ok

{
"contents":[{
"id":0,
"name":"C",
"type":"Directory",
"children":[{
"id":1,
"name":
"acct_locked.html",
"type":"File"
},
{
"id":2,
"name":"certfailure.html",
"type":"File"
},
{
...
}
}]
}]
}
```

Retrieving the contents of a file in the administration pages root with web service:

To do this with the RESTful web service, issue an HTTP GET command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The path of the file to be retrieved.

Response

HTTP response code and contents of the file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Response:

```
200 ok

{"contents":"The contents of the file"}
```

Creating a file in the administration pages root with web service:

To create a file in the administration pages root with the RESTful web service, issue an HTTP POST command on the Administration Pages Root files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The relative path in which the file is created. The top-level directory must be one of `management`, `errors`, `oauth`, or `junction-root`. For example "`management/C`".
name	The name for the new file.

Parameter	Description
type	The type of resource to create. In this case, it is "file".
contents	Optionally the contents of the new file can be passed along using this parameter. If this parameter is not present, then an empty file is created.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}

POST_DATA: {
"name": "new file name",
"type":"file",
"contents":"optional file contents"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Creating a directory in the administration pages root with web service:

To create a directory in the administration pages root with the RESTful web service, issue an HTTP POST command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root/{id}
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in “Configuration changes commit process” on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Parameter	Description
id	The relative path in which the directory is created. The top-level directory must be one of `management`, `errors`, `oauth` or `junction-root`. For example "`management/C`".
name	The name for the new directory.
type	The type of resource to create. In this case, it is "directory".

Response

HTTP response code and JSON data that represents the new directory name.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root/{id}

POST_DATA: {
"name": "new directory name",
"type":"directory"
}
```

Response:

```
200 ok

{"id":"new directory name"}
```

Importing a file to the administration pages root with web service:

To import a file to the administration pages root with the RESTful web service, issue an HTTP POST command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root/{id}
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in “Configuration changes commit process” on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Parameter	Description
id	The relative path in which the file is created. The top-level directory must be one of `management`, `errors`, `oauth` or `junction-root`. For example "`management/C`".
file	The actual file to import as an octet stream.
type	The type of resource to create. In this case, the value is “file”.
force	Whether to force an existing file of the same name to be overwritten. The value can be “yes” or “no”. The default value is “no”.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root/{id}

POST_DATA: {
"file":"file_to_import"(as application/octet-stream),
"type":"file",
"force":"yes"
}
```

Response:

```
200 ok
```

Importing the contents of a .zip file to the administration pages root with web service:

To do this with the RESTful web service, issue an HTTP POST command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in “Configuration changes commit process” on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
file	The .zip file that contains the files to import into the management root. **Note:** The **file** parameter is required. The structure of the .zip file must match the structure of the administration pages root (top-level management, error, oauth, and junction-root directories with the various files in those directories).

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{instance_id}
/management_root

POST_DATA: {
"file":"Zipped configuration file to import" (as application/octet-stream)
}
```

Response:

```
200 ok
```

Exporting a file in the administration pages root with web service:

To export the contents of or export a file in the management root with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The relative path of the file to be exported. The top-level directory must be one of `management`, `errors`, `oauth`, or `junction-root`. For example, `management/C`.

Response

HTTP response code and the file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
?export
```

Response:

```
200 ok

The file text
```

Exporting the contents of the administration pages root as a .zip file with web service:

To do this with the RESTful web service, issue an HTTP GET command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Response

HTTP response code and compressed management root file.

Notes:

- For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.
- Redirect the output of the web service to a local file as it is not screen readable.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/management_root?export
```

Response:

```
200 ok

The zipped management root file
```

Renaming a file in the administration pages root with web service:

To rename a file in the administration pages root with the RESTful web service, issue an HTTP PUT command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The relative path of the file to be renamed. The top-level directory must be one of `management`, `errors`, `oauth`, or `junction-root`. For example, `management/C`.
new_name	This is the new file name.
type	The type of resource to rename. In this case, it is "file".

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}

PUT_DATA: {
"new_name": "new file name",
"type":"file"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Renaming a directory in the administration pages root with web service:

To rename a directory in the administration pages root with the RESTful web service, issue an HTTP PUT command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The relative path of the directory to be renamed. The top-level directory must be one of `management`, `errors`, `oauth`, or `junction-root`. For example, `management/C`.
new_name	This is the new directory name.
type	The type of resource to rename. In this case, it is "directory".

Response

HTTP response code and JSON data that represents the new directory name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}

PUT_DATA: {
"new_name": "new directory name",

"type":"directory"
}
```

Response:

```
200 ok

{"id":"new directory name"}
```

Updating a file in the administration pages root with web service:

To update a file in the administration pages root with the RESTful web service, issue an HTTP PUT command on the Administration Pages Root resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The relative path of the file to be updated. The top-level directory must be one of `management`, `errors`, `oauth`, or `junction-root`. For example, `management/C`.
contents	The new contents of the file. Use this parameter when you update the contents of the file directly. When this parameter is used, do not use the **file** parameter at the same time.

Parameter	Description
file	The local file (as an octet stream) that has the contents to update the management file with. This does not trigger a rename to the name of the local file, only the contents is updated. Use this parameter when you update the file by importing the contents of a local file. When this parameter is used, do not use the **contents** parameter at the same time.
type	The type of resource to rename. In this case, the value is "file".

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Update a file directly

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}

PUT_DATA: {
"contents": "new file contents",

"type":"file"
}
```

Update a file by importing the contents of a local file

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}

POST_DATA: {
"file": "File" (as application/octet-stream),

"type":"file"
}
```

Response:

```
200 ok
```

Deleting a file or directory in the administration pages root with web service:

To do this with the RESTful web service, issue an HTTP DELETE command on the Administration Pages Root files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
id	The relative path of the file or directory to be deleted. The top-level directory must be one of `management`, `errors`, `oauth`, or `junction-root`. For example, `management/C`.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/management_root/{id}
```

Response:

```
200 ok
```

Tracing control

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to control tracing.

Trace data is intended primarily for use by IBM Software Support. Trace data might be requested as part of diagnosing a reported problem. However, experienced product administrators can use trace data to diagnose and correct problems in an IBM Security Access Manager environment. See *IBM Security Access Manager Troubleshooting Guide* for more information about trace event logging.

Note: Use trace with caution. It is intended as a tool to use under the direction of IBM Software Support. Messages from tracing are sometimes cryptic, are not translated, and can severely degrade system performance.

Retrieving all trace components and status with web service:

To retrieve all trace components and status with the RESTful web service, issue an HTTP GET command on the Reverse Proxy tracing resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Response

HTTP response code and JSON data that represens an Array of Hashes, where each hash contains key and value pairs for:

Parameter	Description
id	The trace component name
level	The current tracing level for this component. Valid values are integers 0 - 9, where 0 indicates that tracing is not enabled for the component.
file_size	The current accumulated size of the trace file and all rollover trace files for this component
flush_interval	The number of seconds between the flushing of cached records to the log file
rollover_size	The maximum file size (in bytes) before the file is rolled-over

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
```

Response:

```
[
{
"id": "pdweb.snoop",
"level": "9",
"file_size": "512",
"flush_interval":"30",
"rollover_size":"1024000"
},
{
"id": "pdweb2.snoop",
"level": "5",
"file_size": "1024",
"flush_interval":"40",
"rollover_size":"9999"
},
{
....
}
]
```

Retrieving a snippet of a trace file for a component with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the Reverse Proxy tracing files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the relevant component.
file_id	ID of the relevant file.
options=line-numbers	This parameter ensures that the returned log file snippet includes line numbers.
size="*snippet_size*"	This parameter determines the number of lines to be returned if the default size of 100 lines is not required. **Note:** The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
start="*starting_line_number*"	This parameter determines the line number in the log file where the snippet starts. If it is not set, then the snippet is retrieved from the end of the log file.

Response

HTML response code and JSON data that represents a snippet of the trace file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Retrieve the default snippet size from the tail of the log file

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}
```

Retrieve the default snippet size with line numbers appended to the output

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}?options=line-numbers
```

Retrieve the last 200 lines from the tail of the log file

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}?size=200
```

Retrieve 200 lines starting from line number 1000 of the log file

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}?size=200&start=1000
```

Response:

```
200 ok

{"contents":"The snippet of the trace file"}
```

Listing trace files for a particular component with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the Reverse Proxy tracing files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
/{component_id}/trace_files
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the tracing component to be updated.

Response

HTTP response code and JSON data that represents the trace file names, file sizes, and version information.

Parameter	Description
id	The trace file name.
file_size	The current size of the trace file.
version	The current version of file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
/{component_id}/trace_files
```

Response:

```
[
{
"id":"file_1",
"file_size":"512",
"version":"version_id"
},
{
"id":"file_2",
"file_size":"1024",
"version":"version_id"
},
...
]
```

Modifying the trace level, flush interval, and rollover size for a component with web service:

To complete this operation with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy tracing resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}
/tracing/{component_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the tracing component to be updated.
level	This is the new trace level for the trace component. Valid values are integers 0 - 9, where 0 indicates that tracing is not enabled for the component. This parameter is required.
flush_interval	The number of seconds between the flushing of cached records to the log file. This parameter is optional. If not specified, then the default flush interval is set.
rollover_size	The maximum file size (in bytes) before the file is rolled-over. This parameter is optional. If not specified, then the default rollover size is set.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}
/tracing/{component_id}

PUT_DATA: {
"level": "9",
"flush_interval":"512",
"rollover_size":"9999"
}
```

Response:

```
200 ok
```

Exporting the trace file or rollover trace file for a component with web service:

To export the trace file or rollover trace file for a component with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the Reverse Proxy tracing files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the relevant component.
file_id	ID of the relevant file.

Response

HTTP response code and the trace file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}?export
```

Response:

```
200 ok

The trace file text
```

Deleting a trace file for a component with web service:

To delete the trace file for a component with the RESTful web service, issue an HTTP DELETE command on the Reverse Proxy tracing files resource URI. The trace file to be deleted can not be in use at the time of deletion.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/tracing/{component_id}
/trace_files/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: A trace file can be deleted only when it is not actively being used.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the relevant component.
file_id	ID of the relevant file.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
/{component_id}/trace_files/{file_id}
```

Response:

```
200 ok
```

Deleting the trace file and all rollover files for a component with web service:

To delete the trace file and rollover files for a component with the RESTful web service, issue an HTTP DELETE command on the Reverse Proxy tracing files resource URI. The trace file and rollover file to be deleted should not be in use at the time of deletion.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
/{component_id}/trace_files
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: A trace file can be deleted only when it is not actively being used. This operation only deletes the files that are not in use.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the relevant component.
file_id	ID of the relevant file.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/tracing
/{component_id}/trace_files
```

Response:

```
200 ok
```

Logging

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage the reverse proxy log files.

Retrieving the names of all common log files and file sizes with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the common log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/common
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

N/A

Response

HTTP response code and JSON data that represents the log file names, file sizes, and version information.

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The log file name.
file_size	The current size of the log file.
version	The current version of file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy_logging/common
```

Response:

```
[
{
"id": "file_name",
"file_size": "512"
"version":"version_id"
},
{
"id": "file_name2",
"file_size": "1024"
"version":"version_id"
},
{
....
}
]
```

Retrieving the names of all instance-specific log files and file sizes with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the instance log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/instance/{instance_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Response

HTTP response code and JSON data that represents the log file names, file sizes, and version information.

Parameter	Description
id	The log file name
file_size	The current size of the log file
version	The current version of file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy_logging/instance/{instance_id}
```

Response:

```
[
{
"id": "file_name",
"file_size": "512"
"version":"version_id"
},
{
"id": "file_name2",
"file_size": "1024"
"version":"version_id"
},
{
....
}
]
```

Retrieving a snippet of a common log file with web service:

To retrieve a snippet of a common log file with the RESTful web service, issue an HTTP GET command on the common log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/common/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
file_id	ID of the relevant log file.
options=line-numbers	This parameter ensures that the returned log file snippet includes line numbers.
size="*snippet_size*"	This parameter determines the number of lines to be returned if the default size of 100 lines is not required. **Note:** The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
start="*starting_line_number*"	This parameter determines the line number in the log file where the snippet starts. If it is not set, then the snippet is retrieved from the end of the log file.

Response

HTML response code and JSON data that represents the snippet of the log file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

The following command retrieves the default snippet size from the tail of the log file.

```
GET https://{appliance_hostname}/reverseproxy_logging/common/{file_id}
```

The following command retrieves the default snippet size with line numbers appended to the output.

```
GET https://{appliance_hostname}/reverseproxy_logging/common/{file_id}
?options=line-numbers
```

The following command retrieves the last 200 lines from the tail of the log file.

```
GET https://{appliance_hostname}/reverseproxy_logging/common/{file_id}?size=200
```

The following command retrieves 200 lines that start from line number 1000 of the log file.

```
GET https://{appliance_hostname}/reverseproxy_logging/common/{file_id}
?size=200&start=1000
```

Response:

```
200 ok

{"contents":"The snippet of the log file"}
```

Retrieving a snippet of an instance-specific log file with web service:

To retrieve a snippet of an instance-specific log file with the RESTful web service, issue an HTTP GET command on the instance log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/instance/{instance_id}/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
file_id	ID of the relevant file.
options=line-numbers	This parameter ensures that the returned log file snippet includes line numbers.
size="*snippet_size*"	This parameter determines the number of lines to be returned if the default size of 100 lines is not required. **Note:** The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
start="*starting_line_number*"	This parameter determines the line number in the log file where the snippet starts. If it is not set, then the snippet is retrieved from the end of the log file.

Response

HTTP response code and JSON data that represents the snippet of the file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Retrieve the default snippet size from the tail of the log file

```
GET https://{appliance_hostname}/reverseproxy_logging/instance
/{instance_id}/{file_id}
```

Retrieve the default snippet size with line numbers appended to the output

```
GET https://{appliance_hostname}/reverseproxy_logging/instance
/{instance_id}/{file_id}?options=line-numbers
```

Retrieves the last 200 lines from the tail of the log file

```
GET https://{appliance_hostname}/reverseproxy_logging/instance
/{instance_id}/{file_id}?size=200
```

Retrieve 200 lines that start from line number 1000 of the log file

```
GET https://{appliance_hostname}/reverseproxy_logging/instance
/{instance_id}/{file_id}?size=200&start=1000
```

Response:

```
200 ok

{"contents":"The snippet of the log file"}
```

Exporting a common log file with web service:

To export a common log file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the common log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/common/{file_id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
file_id	The relative path to which the log file contents is exported.

Response

HTTP response code and log file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy_logging/common/{file_id}?export
```

Response:

```
200 ok

The log file text
```

Exporting an instance-specific log file with web service:

To export an instance-specific log file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the instance log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/instance/{instance_id}/{file_id}
?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
file_id	The relative path to which the log file contents is exported.

Response

HTTP response code and the log file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy_logging/instance
/{instance_id}/{file_id}?export
```

Response:

```
200 ok

The log file text
```

Clearing a common log file with web service:

To clear a common log file with the RESTful web service, issue an HTTP DELETE command on the common log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/common/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
file_id	The relative path of the log file to be cleared.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy_logging/common/{file_id}
```

Response:

```
200 ok
```

Clearing an instance-specific log file with web service:

To clear an instance-specific log file with the RESTful web service, issue an HTTP DELETE command on the instance log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy_logging/instance/{instance_id}/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
file_id	The relative path of the log file to be cleared.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy_logging/instance/{instance_id}
/{file_id}
```

Response:

```
200 ok
```

Statistics control

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage statistics.

Retrieving all statistics components and details with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the Reverse Proxy statistics resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Response

HTTP response code and JSON data that represents the statistics components and their status.

Parameter	Description
id	The statistics component name.
status	The current statistics status for this component. Can be either on or off.
file_size	The current accumulated size of the statistics log file and all rollover files for this component in bytes.
interval	The time interval between reports of information in the format hours:minutes:seconds.
count	The number of entries that is sent to the log file.
flush_interval	The number of seconds between the flushing of cached records to the log file. This property exists only if the status is on.
rollover size	The maximum file size (in bytes) before the file is rolled-over. This property exists only if the status is on.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
```

Response:

```
[{
"id": "statistics_component",
"status": "on/off",
"file_size": "512",
"interval":"interval",
"count":"number_of_entries",
"flush_interval":"flush_interval",
"rollover_size":"rollover_size"
},
{
"id": "statistics_component2",
"status": "on/off",
"file_size": "1024",
"interval":"interval",
"count":"number_of_entries",
"flush_interval":"flush_interval",
"rollover_size":"rollover_size"
},
{
....
}]
```

Retrieving all statistics log file names and file sizes for a component with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the Reverse Proxy statistics log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	The relative path of the statistics component.

Response

HTTP response code and JSON data that represents the statistics log files, file sizes, and version information.

Parameter	Description
id	The statistics log file name.
file_size	The current size of the file in bytes.
version	The current version of file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files
```

Response:

```
[
{
"id": "file_name",
"file_size": "512",
"version":"version_id"
},
{
"id": "file_name2",
"file_size": "1024",
"version":"version_id"
},
{
....
}
]
```

Retrieving a snippet of a statistics log file for a component with web service:

To retrieve a snippet of a statistics log file for a component with the RESTful web service, issue an HTTP GET command on the Reverse Proxy statistics log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the statistics component.
file_id	The relative path of the statistics log file.
options=line-numbers	This parameter ensures that the returned log file snippet includes line numbers.
size="*snippet_size*"	This parameter determines the number of lines to be returned if the default size of 100 lines is not required. **Note:** The maximum size that can be returned is 214800000 lines. If a size greater than that is specified, then the maximum (214800000 lines) is returned.
start="*starting_line_number*"	This parameter determines the line number in the log file where the snippet starts. If it is not set, then the snippet is retrieved from the end of the log file.

Response

HTTP response code and JSON data representing the snippet of the log file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Retrieve the default snippet size from the tail of the log file

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}
```

Retrieve the default snippet size with line numbers appended to the output

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}?options=line-numbers
```

Retrieve the last 200 lines from the tail of the log file

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}?size=200
```

Retrieve 200 lines that start from line number 1000 of the log file

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}?size=200&start=1000
```

Response:

```
200 ok

{"contents":"The snippet of the log file"}
```

Exporting the statistics log file or rollover statistics log file for a component with web service:

To export the statistics log file or rollover statistics log file for a component to the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the Reverse Proxy statistics log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the statistics component.
file_id	The relative path to which the statistics log file is exported.

Note: The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Response

HTTP response code and the log file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}?export
```

Response:

```
200 ok

Log file text
```

Modifying statistics settings for a component with web service:

To modify the statistics settings for a component with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy statistics resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics/{component_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	The relative path of the statistics component.
status	The new status for this component. Can be either "on" or "off". If the value is set to "off", all other parameters are ignored.
interval	The time interval between reports of information in the format hours:minutes:seconds or minutes:seconds or just seconds. This parameter is not required if any of interval_hours, interval_mins or interval_secs are used. If this parameter is used, the value must be quoted and escaped. For example, `\"hours:mins:secs\"`.
interval_hours	The time interval between reports of information in hours. This parameter can be used with interval_mins and interval_secs and its usage means that the interval parameter is not used.
interval_mins	The time interval between reports of information in minutes. This parameter can be used with interval_hours and interval_secs and its usage means that the interval parameter is not used.
interval_secs	The time interval between reports of information in seconds. This parameter can be used with interval_hours and interval_mins and its usage means that the interval parameter is not used.
count	The number of entries that is be sent to the log file.
flush_interval	The number of seconds between the flushing of cached records to the log file.
rollover_size	The maximum file size (in bytes) before the file is rolled-over.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Modifying statistics settings specifying the interval parameter

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}
/statistics/{component_id}

PUT_DATA: {
"status": "on/off",
"interval":"interval",
"count":"number_of_entries",
"flush_interval":"flush_interval",
"rollover_size":"rollover_size"
}
```

Modifying statistics settings specifying the interval hours, minutes, and seconds parameters

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}
/statistics/{component_id}

PUT_DATA: {
"status": "on/off",
"interval_hours":"interval_hours",
"interval_mins":"interval_mins",
"interval_secs":"interval_secs",
"count":"number_of_entries",
"flush_interval":"flush_interval",
"rollover_size":"rollover_size"
}
```

Response:

```
200 ok
```

Deleting the statistics log file or rollover file for a component with the web service:

To complete this operation with the RESTful web service, issue an HTTP DELETE command on the Reverse Proxy statistics log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: Only statistics log files that are no longer in use can be deleted.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the statistics component.

Parameter	Description
file_id	The relative path of the file to be deleted.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files/{file_id}
```

Response:

```
200 ok
```

Deleting the unused statistics log file and rollover files for a component with web service:

To complete this operation with the RESTful web service, issue an HTTP DELETE command on the Reverse Proxy statistics log files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the statistics component.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/statistics
/{component_id}/stats_files
```

Response:

```
200 ok
```

Message routing control

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to control message routing.

Retrieving a routing control file with web service:

To retrieve routing control file with the RESTful web service, issue an HTTP GET command on the Reverse Proxy configuration routing resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Response

HTTP response code and JSON data that represents the file contents.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file
```

Response:

```
200 ok

{"contents":"The routing file contents"}
```

Exporting a routing control file with web service:

To export a routing control file with the RESTful web service, issue an HTTP GET command with the `export` parameter on the Reverse Proxy configuration routing resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Note: The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, the content of the file is retrieved.

Response

HTTP response code and the routing file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file?export
```

Response:

```
200 ok

The routing file text
```

Updating routing control file data with web service:

To update a routing control file with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy configuration routing resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	The instance name to import file to.
file_contents	The file content to be imported as application or octet-stream.
file	The file that contains the new data as application/octet-stream.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Pass the new file contents as JSON

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file

PUT_DATA: {
"file_contents": "file contents to import"
}
```

Pass the new file contents as a file

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}/routing_file

PUT_DATA: {
"file": The file containing the new data as application/octet-stream
}
```

Response:

```
200 ok
```

Transaction logging

You can manage transaction logging with the Reverse Proxy web service.

Retrieving all transaction logging components and their details with web service:

To complete this operation with the RESTful web service, issue an HTTP GET command on the Reverse Proxy transaction logging resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.

Response

HTTP response code and JSON data that represent the details of transaction logging components.

Parameter	Description
id	The transaction logging component name.
status	The status for this component. Can be either “on” or “off”.
rollover_size	The maximum file size of the transaction logging data file for this component in bytes before it is rolled over.
file_size	The current accumulated size of the transaction logging data file and all rollover files for this component in bytes.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
```

Response:

```
200 ok

[
{
"id": "transaction_component",
"status": "on/off",
"rollover_size": "512000",
"file_size":"512"
},
{
"id": "transaction_component2",
"status": "on/off",
"rollover_size": "1024000",
"file_size":"1024"
},
{
....
}
]
```

Retrieving all transaction logging data file names and file sizes for a component with web service:

To complete the operation with the RESTful web service, issue an HTTP GET command on the Reverse Proxy transaction logging resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the transaction logging component.

Response

HTTP response code and JSON data that represent the transaction logging data file names, file sizes, and version information.

Parameter	Description
id	The transaction logging data file name.
file_size	The current size of the file in bytes.
version	The current version of file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files
```

Response:

```
200 ok

[
{
"id": "file_name",
"file_size": "512",
"version":"version_id"
},
{
"id": "file_name2",
"file_size": "1024",
"version":"version_id"
},
```

```
{
....
}
]
```

Rolling over the transaction logging data file for a component with web service:

To complete this operation with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy transaction logging data files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}
/transaction_logging/{component_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the transaction logging component.
rollover	An indication that the operation required is a data file rollover. In this case, the value is be "yes".

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}
/transaction_logging/{component_id}

PUT_DATA: {
"rollover": "yes"
}
```

Response:

```
200 ok
```

Exporting the transaction logging data file or rollover transaction logging data file for a component with web service:

To complete the operation with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the Reverse Proxy transaction logging data files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files/{file_id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. The difference is that the export response is the text of the certificate request, while the retrieve response is the certificate request text in JSON format.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the transaction logging component.
file_id	The relative path to which the transaction logging data file or rollover transaction logging data file is exported.

Response

HTTP response code and the transaction log file.

Notes:

- For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.
- The output of the web service must be redirected to a local file as it is not screen readable.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files/{file_id}?export
```

Response:

```
200 ok

The transaction log text
```

Modifying the status and rollover size for a component with web service:

To complete the operation with the RESTful web service, issue an HTTP PUT command on the Reverse Proxy transaction logging resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}
/transaction_logging/{component_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the transaction logging component to be updated.
status	The new status for this component. Can be either "on" or "off".
rollover_size	The maximum data file size in bytes before the file is rolled over.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{instance_id}
/transaction_logging/{component_id}

PUT_DATA: {
"status": "on/off",
"rollover_size":"size in bytes"
}
```

Response:

```
200 ok
```

Deleting the transaction logging data file or rollover file for a component with web service:

To complete the operation with the RESTful web service, issue an HTTP DELETE command on the Reverse Proxy transaction logging data files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files/{file_id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: Only transaction logging data files that are no longer in use can be deleted.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the transaction logging component.
file_id	The relative path of the file to be deleted.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files/{file_id}
```

Response:

```
200 ok
```

Deleting the unused transaction logging data file and rollover files for a component with web service:

To complete the operation with the RESTful web service, issue an HTTP DELETE command on the Reverse Proxy transaction logging data files resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
instance_id	ID of the relevant instance.
component_id	ID of the transaction logging component.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{instance_id}/transaction_logging
/{component_id}/translog_files
```

Response:

```
200 ok
```

Junction management

Use WebSEAL Instances resource URI to manage standard and virtual junctions.

Retrieving a list of standard and virtual junctions with web service:

To retrieve a list of standard and virtual junctions with the web service, issue an HTTP GET command on the WebSEAL Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The WebSEAL instance name.

Response

HTTP response code and JSON data that represents the list of junctions.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
```

Response:

```
200 ok

[
{"id":"junction point","type":"junction type"},
{"id":"junction point","type":"junction type"},
...
]
```

Retrieving the parameters for a single standard or virtual junction with web service:

To complete the operation with the RESTful web service, issue an HTTP GET command on the WebSEAL Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
?junctions_id={junctionname}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The WebSEAL instance name.
junctions_id	The name of the standard or virtual junction to retrieve parameters for.

Response

HTTP response code and JSON representing the parameters of the junction.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
?junctions_id={junctionname}
```

Response:

```
200 ok

[
{"junction_point":"testvirtual",
"junction_type":"TCP",
"junction_hard_limit":"0 - using global value",
"junction_soft_limit":"0 - using global value",
...
"servers":[{"server_uuid":"02a53f5e-cab2-11e1-9ace-000c29411c35","server_state
":"running","operation_state":"Online","server_hostname":"w3.ibm.com","server_po
rt":"80","server_dn":"","local_ip":"","query_content_url":"/cgi-bin/query_conten
```

```
ts","query_contents":"unknown","case_sensitive_url":"no","windows_style_url":"ye
s","current_requests":"0","total_requests":"8"}
]
]}
```


Creating a standard or virtual junction with web service:

To create a standard or virtual junction with the RESTful web service, issue an HTTP POST command on the WebSEAL Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The WebSEAL instance name.
server_hostname	The DNS host name or IP address of the target back-end server. This parameter is required.
junction_point	Name of the location in the WebSEAL namespace where the root of the back-end application server namespace is mounted. For standard junctions, the value is "/" followed by the junction name. For example, `/test` For virtual junctions, the value is the junction name only. For example, `test` This parameter is required.
junction_type	Type of junction. The value is one of: **tcp**, **ssl**, **tcpproxy**, **sslproxy**, **local**, **mutual** Default port for **–t tcp** is 80. Default port for **–t ssl** is 443. This parameter is required.
junction_hard_limit	Defines the hard limit percentage for consumption of worker threads. Valid value is an integer from "0" to "100".
junction_soft_limit	Defines the soft limit percentage for consumption of worker threads. Valid value is an integer from "0" to "100".

Parameter	Description
basic_auth_mode	Defines how the WebSEAL server passes client identity information in HTTP basic authentication (BA) headers to the back-end server. The value is one of: **filter** (default), **ignore**, **supply**, **gso**.
tfim_sso	Enables IBM Security Federated Identity Manager single-signon (SSO) for the junction. Valid value is “yes” or “no”.
remote_http_header	Controls the insertion of Security Access Manager for Web-specific client identity information in HTTP headers across the junction. The value is one of: **iv-user**, **iv-user-l**, **iv-groups**, **iv-creds**, **all**.
stateful_junction	Specifies whether the junction support stateful applications. By default, junctions are not stateful. Valid value is “yes” or “no”.
preserve_cookie	Specifies whether modifications of the names of non-domain cookies are to be made. Valid value is “yes” or “no”. This parameter is not valid for virtual junctions.
cookie_include_path	Specifies whether script generated server-relative URLs are included in cookies for junction identification. Valid value is “yes” or “no”. This parameter is not valid for virtual junctions.
transparent_path _junction	Specifies whether a transparent path junction is created. Valid value is “yes” or “no”. This parameter is not valid for virtual junctions.
mutual_auth	Specifies whether to enforce mutual authentication between a front-end WebSEAL server and a back-end WebSEAL server over SSL. Valid value is “yes” or “no”.
insert_ltpa_cookies	Controls whether LTPA cookies are passed to the junctioned Web server. Valid value is “yes” or “no”.
insert_session_cookies	Controls whether to send the session cookie to the junctioned Web server. Valid value is “yes” or “no”.
request_encoding	Specifies the encoding to use when the system generates HTTP headers for junctions. Possible values for encoding are: **utf8_bin**, **utf8_uri**, **lcp_bin**, and **lcp_uri**.

Parameter	Description
enable_basic_auth	Specifies whether to use BA header information to authenticate to back-end server. Valid value is “yes” or “no”.
key_label	The key label for the client-side certificate that is used when the system authenticates to the junctioned Web server.
gso_resource_group	The name of the GSO resource or resource group.
junction_cookie _javascript_block	Controls the junction cookie JavaScript block. The value should be one of: **trailer**, **inhead**, **onfocus**, **xhtml10**. • Use **trailer** to append the junction cookie JavaScript to HTML page returned from back-end server. • Use **inhead** to insert the JavaScript block between <header> </header> tags for HTML 4.01 compliance. • Use **onfocus** to use the onfocus event handler in the JavaScript to ensure the correct junction cookie is used in a multiple-junction/multiple-browser-window scenario. • Use **xhtml10** to insert a JavaScript block that is HTML 4.01 and XHTML 1.0 compliant.
set_cookie_header	Whether to add the original path attribute value (for example, /orders) to the modified name of the cookie to ensure that the set-cookie header is unique. Valid value is “yes” or “no”.
client_ip_http	Specifies whether to insert the IP address of the incoming request into an HTTP header for transmission to the junctioned Web server. Valid value is “yes” or “no”.
version_two_cookies	Specifies whether LTPA version 2 cookies (LtpaToken2) are used. Valid value is “yes” or “no”.
ltpa_keyfile	Location of the key file that is used to encrypt the LTPA cookie data.
authz_rules	Specifies whether to allow denied requests and failure reason information from authorization rules to be sent in the Boolean Rule header (AM_AZN_FAILURE) across the junction. Valid value is “yes” or “no”.
fsso_config_file	The name of the configuration file that is used for forms based single sign-on.
username	WebSEAL user name. Used to send BA header information to the backend server.
password	WebSEAL password. Used to send BA header information to the backend server.

Parameter	Description
server_uuid	Specifies the UUID that will be used to identify the junctioned Web server. This field is used for stateful junctions.
server_port	TCP port of the back-end third-party server. Default is 80 for TCP junctions and 443 for SSL junctions.
virtual_hostname	Virtual hostname that is used for the junctioned Web server.
server_dn	Specifies the distinguished name of the junctioned Web server.
local_ip	Specifies the local IP address that WebSEAL uses when the system communicaes with the target back-end server. If this option is not provided, WebSEAL uses the default address as determined by the operating system. If you supply an address for a particular junction, WebSEAL is modified to bind to this local address for all communication with the junctioned server.
query_contents	Provides WebSEAL with the correct name of the **query_contents** program file and where to find the file. By default, the Windows file is called `query_contents.exe` and the UNIX file is called `query_contents.sh`. By default, WebSEAL looks for the file in the cgi_bin directory of the back-end Web server. Required for back-end Windows and UNIX Web servers
case_sensitive_url	Specifies whether the Web Reverse Proxy server treats URLs as case sensitive. Valid value is “yes” or “no”.
windows_style_url	Specifies whether Windows style URLs are supported. Valid value is “yes” or “no”.
ltpa_keyfile_password	Password for the key file that is used to encrypt LTPA cookie data.
https_port	HTTPS port of the back-end third-party server. Applicable when the junction type is **ssl**.
http_port	HTTP port of the back-end third-party server. Applicable when the junction type is **tcp**.
proxy_hostname	The DNS host name or IP address of the proxy server. Applicable when the junction type is **sslproxy**.
proxy_port	The TCP port of the proxy server. Applicable when the junction type is **tcpproxy**.
sms_environment	Only applicable to virtual junctions. Specifies the replica set that sessions on the virtual host junction are managed under. The replica set also provide capability to group or separate log in sessions among multiple virtual hosts. If this option is not used to specify the replica set for the virtual host junction, the virtual host junction is automatically assigned to a replica set matching its virtual host name. For example, if the virtual host name is `vhostA.example.com`, the replica set is named `vhostA.example.com`. The replica set used for the virtual host junction must be present in the `[replica-sets]` stanza of the WebSEAL configuration file. This option cannot be used in a non-SMS environment.

Parameter	Description
vhost_label	Only applicable to virtual junctions. Causes a second additional virtual host junction to share protected object space as the initial virtual host junction. This option is appropriate for junction pairs only (2 junctions with complementary protocols). The option does not support the association of more than 2 junctions.
force	Specifies whether to overwrite an existing junction of the same name. The value is “true” or “false”.

Response

HTTP response code and JSON data that represents result of the operation.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
POST https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions

POST_DATA: {
"junction_point":"/test",
"junction_type":"...",
"junction_hard_limit":"...",
...
"server_uuid":"...",
"server_state":"...",
...
}
```

Response:

```
200 ok

{"id":"The new junction point",
"message":"Any messages generated when creating the junction"}
```

Adding a back-end server to an existing standard or virtual junctions with web service:

To complete this operation with the RESTful web service, issue an HTTP PUT command on the WebSEAL Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The Web Reverse Proxy instance name.
server_hostname	The DNS host name or IP address of the target back-end server. This parameter is required.
junction_point	Name of the location in the WebSEAL namespace where the root of the back-end application server namespace is mounted. For standard junctions, the value is "/" followed by the junction name. For example, `/test` For virtual junctions, the value is the junction name only. For example, `test` This parameter is required.
junction_type	Type of junction. The value is one of: **tcp**, **ssl**, **tcpproxy**, **sslproxy**, **local**, **mutual** Default port for **–t tcp** is 80. Default port for **–t ssl** is 443. This parameter is required.
server_port	TCP port of the back-end third-party server. Default is 80 for TCP junctions and 443 for SSL junctions.
virtual_hostname	Virtual host name that is used for the junctioned Web server.
virtual_https_hostname	Virtual HTTPS host name that is used for the junctioned Web server.
server_dn	Specifies the distinguished name of the junctioned Web server.
query_contents	Provides WebSEAL with the correct name of the **query_contents** program file and where to find the file. By default, the Windows file is called `query_contents.exe` and the UNIX file is called `query_contents.sh`. By default, WebSEAL looks for the file in the cgi_bin directory of the back-end Web server. Required for back-end Windows and UNIX Web servers.
stateful_junction	Specifies whether the junction support stateful applications. By default, junctions are not stateful. Valid value is “yes” or “no”.
case_sensitive_url	Specifies whether the Web Reverse Proxy server treats URLs as case sensitive. Valid value is “yes” or “no”.
windows_style_url	Specifies whether Windows style URLs are supported. Valid value is “yes” or “no”.
https_port	HTTPS port of the back-end third-party server. Applicable when the junction type is **ssl**.
http_port	HTTP port of the back-end third-party server. Applicable when the junction type is **tcp**.
proxy_hostname	The DNS host name or IP address of the proxy server. Applicable when the junction type is **sslproxy**.

Parameter	Description
proxy_port	The TCP port of the proxy server. Applicable when the junction type is **tcpproxy**.
sms_environment	Only applicable to virtual junctions. Specifies the replica set that sessions on the virtual host junction are managed under. The replica set also provides the ability to group or separate log in sessions among multiple virtual hosts. If this option is not used to specify the replica set, the virtual host junction is automatically assigned to a replica set matching its virtual host name. For example, if the virtual host name is `vhostA.example.com`, the replica set is named `vhostA.example.com`. The replica set used for the virtual host junction must be present in the `[replica-sets]` stanza of the WebSEAL configuration file. This option cannot be used in a non-SMS environment.
vhost_label	Only applicable to virtual junctions. Causes a second additional virtual host junction to share protected object space as the initial virtual host junction. This option is appropriate for junction pairs only (2 junctions with complementary protocols). The option does not support the association of more than 2 junctions.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions

PUT_DATA: {
"junction_point":"/test",
"server_uuid":"...",
"server_state":"...",
...
}
```

Response:

```
200 ok
```

Deleting a standard or virtual junction with web service:

To delete a standard or virtual junction with the RESTful web service, issue an HTTP DELETE command on the WebSEAL Instances resource URI.

URL

```
https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
?junctions_id=/{junctionname}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
reverseproxy_id	The WebSEAL instance name.
junctions_id	The name of the standard or virtual junction to delete.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/reverseproxy/{reverseproxy_id}/junctions
?junctions_id=/{junctionname}
```

Response:

```
200 ok
```

Web application firewall with PAM engine

The web application firewall monitors and potentially blocks access from a web application if it does not meet the configured policy of the firewall. It uses the IBM Security Systems Protocol Analysis Module (PAM) as its engine.

It controls the input, output, and access from a web application. Because it acts on the application layer, it can inspect the contents of the traffic and block specified content. Such content includes certain websites, viruses, attempts to use known logical flaws in web applications.

Deep packet inspection:

The Web Gateway Appliance uses the IBM Security Systems Protocol Analysis Module (PAM) to perform deep packet inspection. It is the same engine as the engine used in the IBM Intrusion Prevention Appliance. It checks the content of packets for threats and takes actions against the threat according to configured policies.

When a threat is found, the appliance can take one of the following actions:

1. Generate an audit event.
2. Drop connection from the client. Generate an audit event. (With this option, the client is set to be blocked for 0 seconds.)
3. Drop connection from the client. Block the client from accessing for a pre-defined period. Generate an audit event.

PAM uses content recognition to identify the file format for most of the files that it inspects. For example, if a file begins with "GIF89a", it is almost certainly a GIF image format file. (Even if it is not actually a GIF formatted image, most applications assume that it is and attempt to parse it as such.)

If PAM cannot recognize the file format based upon its content, it evaluates any stateful information about the file that the client and server previously negotiated. That is, it determines the file format by using the MIME type and extension (".exe" for example) on the file name.

Finally, if PAM cannot recognize the content and no stateful information is available, it determines a file format based upon the protocol that is used to transfer the file. For example, if HTTP is used to transfer the file, PAM assumes that the file contains HTML data.

For more details about the threats that PAM can detect and the configuration parameters of the PAM engine, see the PAM documentation. It is available for download at http://www.iss.net/security_center/reference/help/pam

Note: The download link points to a .zip file that contains the PAM documentation in .chm format. You can view it on Windows. If you are using an operating system other than Windows, you might need an additional program to open the .chm file.

You can also view references about a particular issue at http://www.iss.net/security_center/reference/vuln/<*ISSUE_NAME*>.htm

For example, to view details about the issue of SQL injection, go to http://www.iss.net/security_center/reference/vuln/SQL_Injection.htm

Configuring the web application firewall with web service:

To configure Web Application Firewall with the RESTful web service, add the following stanza and entries to the Reverse Proxy configuration file using the Configuration Entry Management web service.

Example

```
# The PAM stanza is used to house the configuration data which
# is required for the PAM integration. The PAM functionality
# is used to provide deep content packet inspection on selected
# requests, checking for potential security vulnerabilities.
#

[PAM]

#
# Whether PAM analysis is enabled.
#
pam-enabled = false

#
# The name of the PAM distribution directory.
#
pam-distribution-directory = /var/pam

#
# The amount of memory, in bytes, which can be consumed by
# PAM. This allows PAM to tune the size of its caches for the
# amount of available memory.
#
pam-max-memory = 16777216

#
# The following item controls whether the X-Forwarded-For header
# is used to identify the client.  This configuration item is useful
# if a network terminating proxy is sitting between the server and the
```

```
# client.  If the value is set to false the client will be identified
# based on the socket connection information.
#
pam-use-proxy-header = false

#
# Any specific parameters which should be passed to the PAM
# HTTP interface during initialization. Refer to the PAM
# documentation for a list of valid PAM parameters.
#
# The configuration entry may be specified multiple times,
# one for each PAM parameter. The entry should be of the
# format:
# pam-http-parameter = <parameter>:<value>


#
# Any specific parameters which should be passed to the PAM
# coalescer interface. This interface is used to combine
# related PAM issues into a single event. Refer to the PAM
# documentation for a list of valid parameters.
#
# The configuration entry may be specified multiple times,
# one for each coalescer parameter. The entry should be of
# the format:
# pam-coalescer-parameter = <parameter>:<value>
#
# For example:
# pam-coalescer-parameter = combine:on

#
# The path to our logging directory, used when we have been
# configured with a 'file' logging agent.
#
pam-log-path = /var/pdweb/log

#
# The logging configuration.  The logging configuration consists
# of an agent identifier, followed by attributes which are
# associated with the agent.  Each attribute consists of a
# name/value pair, separated by '=', and each attribute is
# separated by ','.
#
# For example, to configure the auditing of records to a file:
#     file path=pam.log,flush_interval=20,rollover_size=2000000
#
pam-log-cfg = file path=pam.log,flush_interval=20,rollover_size=2000000

#
# Should the audit events be sent to the PAM log file?
# It is worth noting that the number of logged events
# will increase dramatically if this option is enabled.
#
pam-log-audit-events = false

#
# Define which PAM issues will be enabled/disabled.
# The configuration entry is a comma separated list, and each issue
# is enabled or disabled in order.
#
# format:
#    [+-]<issue id>
#
# where:
#     +          : will enable the issue
#     -          : will disable the issue
#     <issue id> : the PAM issue identifier, or 'all' to
```

```
#                  indicate all issues.
#
# For example:
#   to disable all issues but Ace_Filename_Overflow:
#     pam-issues = -all,+2121050
#
#   to enable all issues but HTTPS_Apache_ClearText_DoS:
#     pam-issues = +all,-2114033
#
# By default all issues will be enabled.
pam-issues = +all                                                    #

# The rules which should be applied to determine whether
# a particular resource should be passed down to the PAM
# layer or not. Each rule will be examined in sequence
# until a match is found. The first successful match
# will determine whether the request is passed to the
# PAM layer or not. The request will not be passed to
# the PAM layer if no match is found.
#
# Multiple entries may be specified, and each entry
# should be of the format:
# pam-resource-rule = [+-]{uri}
#
# where:
#   +    : Indicates that matching requests should be
#          passed to the PAM layer.
#   -    : Indicates that matching requests should not
#          be passed to the PAM layer.
#  {uri} : Contains a pattern which is used to match
#          against the URI which is found in the
#          request. The wildcard characters '*'
#          and '?' may be used.
#
# For example:
# pam-resource-rule = -*.gif
# pam-resource-rule = +*.html
#

#
# The following stanza can be used to customize the
# PAM processing for individual resources and events.
# The name of the stanza should be of the format:
# [pam-resource:{uri}]
#
# where:
#   {uri} : Contains a pattern which is used to match
#           against the URI which is found in the
#           request. The wildcard characters '*' and
#           '?' may be used.
#
# For example:
# [pam-resource:*.js]
#

[pam-resource:test.html]

#
# The entries contained within this stanza are used
# to control the processing of certain PAM related
# events. Each entry will be of the format:
# {pam-issue} = {action}
#
# where:
#   {pam-issue} : Contains a pattern which is used to
#                 match a PAM issue. The wildcard
#                 characters '*' and '?' may be
```

```
#                 used.
#   {action}    : The action which is to be undertaken
#                 for the issue. The action can be
#                 one of the following:
#                 - block: Blocks the connection for
#                   a specified number of seconds,
#                   e.g. block:30;
#                 - ignore: Ignore the issue and
#                   continue to process the request;
#                 - drop : Drop the connection.
#
# For example:
# 212105? = drop
# 2119002 = block:20
```

For references about how to use the Configuration Entry Management web service, see the following topics:

- "Retrieving a list of stanzas with web service" on page 83
- "Adding a configuration stanza name with web service" on page 83
- "Adding a configuration entry or entries by stanza with web service" on page 84
- "Updating a configuration entry or entries by stanza with web service" on page 85

Audit format:

There are two audit formats: messages and events. Both messages and events are in XML format. They can be sent to either a log file or a remote server through the PAM auditing system.

Messages

The PAM-related messages are in the following format:

```
<message time="[secs-since-epoch]" value="[message text]"/>
```

For example:

```
<message time="1329864833" value="PAM base=0x7f44b87f6280 guard=0x17810e06
label='PAM2_DAILYBUILD-2012-02-03'"/>
```

Events

The PAM-related events are in the following format:

```
<issue id="[id]" name="[name] start="[time]" intruder="[addr]" victim="[addr]">
     <events agent="[name]" user="[name]" action="[response]">
       <event resource="[url]"/>
     </events>
</issue>
```

Each issue can contain one or more events. The 'agent', 'user' and 'name' fields that are contained within the 'events' construct represent the default values for the contained events. Different values can however be supplied as a part of the 'event' construct.

For example:

```
<issue id="2114033" name="HTTPS_Apache_ClearText_DoS" start="1329864907"
intruder="10.251.140.49:53395" victim="10.251.140.1:443">
    <events action="Drop>
        <event resource="/cgi-bin/dfire.cgi?loop=1"/>
```

```
        <event resource="/cgi-bin/dfire.cgi?loop=1" user="sec_master"
        agent="curl/7.19.7"/>
    </events>
</issue>
```

Dynamic URL (DynURL) configuration file management

Use either the local management interface or web service to manage Dynamic URL (DynURL) configuration files.

Managing DynURL configuration files with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.

Retrieving a list of all DynURL configuration files with local management interface

To retrieve a list of all DynURL configuration files with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. The file name, file size, and last modified time information of all DynURL configuration files is displayed.

Viewing a DynURL configuration file with local management interface

To view the contents of a DynURL configuration files with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Select the file that you want to view.
3. Click **Edit**. The contents of the file is displayed in the pop-up window.

Creating a DynURL configuration file with local management interface

To create a DynURL configuration file with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Click **New**.
3. In the window that pops up:
   a. Modify the content of the file.
   b. Enter the name for the file to be created in the **DynURL Configuration File Name** field.
   c. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing a new DynURL configuration file with local management interface

To import a new DynURL configuration file with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Click **Manage** > **Import**.
3. In the window that pops up, click **Browse**.
4. Choose the file to import from your local workstation.
5. Click **Import**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting a DynURL configuration file with local management interface

To export a DynURL configuration file with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Select the DynURL configuration file to export.
3. Click **Manage** > **Export**.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. In the window that pops up, confirm to save the file to your local workstation.

Updating an existing DynURL configuration file with local management interface

To update an existing DynURL configuration file with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Select the DynURL configuration file to edit.
3. Click **Edit**.
4. In the window that pops up:
   a. Modify the file content.
   b. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Renaming a DynURL configuration file with local management interface

To rename a DynURL configuration file with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Select the DynURL configuration file to rename.
3. Click **Manage** > **Rename**.
4. In the window that pops up, enter the new name for the file in the **New Resource Name** field.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting a DynURL configuration file with local management interface

To delete a DynURL configuration file with the local management interface, use the URL Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **URL Mapping**.
2. Select the DynURL configuration file to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing DynURL configuration files with web service

Use the DynURL Configuration File resource URI to manage DynURL configuration files.

Retrieving a list of all DynURL configuration files with web service

To retrieve a list of all DynURL configuration files with the RESTful web service, issue an HTTP GET command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

N/A

Response

HTTP response code and JSON data that represents the list of DynURL configuration files.

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file.
version	The current version of file. This information is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/dynurl_config
```

Response:

```
200 ok

[
{
"id":"dynurl_config_file_1",
"version":"version_id"
},
{
"id":"dynurl_config_file_2",
"version":"version_id"
},
...
]
```

Retrieving a DynURL configuration file with web service

To retrieve the content of a DynURL configuration file with the RESTful web service, issue an HTTP GET command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to retrieve

Response

HTTP response code and JSON data that represents the file contents.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/dynurl_config/{id}
```

Response:

```
200 ok

{"contents":"The file contents"}
```

Retrieving the DynURL configuration file template with web service

To retrieve the DynURL configuration file template with the RESTful web service, issue an HTTP GET command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/wga_templates/dynurl_template
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the template data.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/dynurl_template
```

Response:

```
200 ok

{"id":"new file name"}
```

Creating a DynURL configuration file with web service

To create a DynURL configuration file with the RESTful web service, issue an HTTP POST command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
name	The new DynURL Configuration file name. This name must be unique.
dynrul_config_data	The data that is saved into the new DynURL configuration file.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/dynurl_config

POST_DATA: {
"dynurl_config_data": "<DATA>"
"name":"new file name"
}
```

Response:

```
200 ok

{"id":"New file name"}
```

Importing a new DynURL configuration file with web service

To import a new DynURL configuration file with the RESTful web service, issue an HTTP POST command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
dynurl_config_file	DynURL configuration file to import

Response

HTTP response code and JSON data that represents the imported file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/dynurl_config

POST_DATA: {
"dynurl_config_file": "DynURL Configuration File to Import"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Exporting a DynURL configuration file with web service

To export a DynURL configuration file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. The difference is that the export response is the text of the certificate request while the retrieve response is the certificate request text in JSON format.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to export

Response

HTTP response code and the file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/dynurl_config/{id}?export
```

Response:

```
200 ok

The file text
```

Exporting the DynURL configuration file template with web service

To export the DynURL configuration file template with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/wga_templates/dynurl_template?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. The difference is that the export response is the text of the certificate request while the retrieve response is the certificate request text in JSON format.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the template file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/dynurl_template?export
```

Response:

```
200 ok

The template file text
```

Updating an existing DynURL configuration file with web service

To update an existing DynURL configuration file with the RESTful web service, issue an HTTP PUT command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The existing DynURL configuration file name.
dynurl_config_data	The data that is saved into the existing DynURL configuration file.
dynurl_config_file	The file that contains the new data to be passed into the existing DynURL configuration file.
mode	The value is always be "update".

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Pass the new file contents as JSON

```
PUT https://{appliance_hostname}/dynurl_config/{id}

PUT_DATA: {
"dynurl_config_data": "file contents to update"
"mode":"update"
}
```

Pass the new file contents as a file

```
PUT https://{appliance_hostname}/dynurl_config/{id}

PUT_DATA: {
"dynurl_config_file": The file containing the new data
as application/octet-stream
"mode":"update"
}
```

Response:

```
200 ok
```

Renaming a DynURL configuration file with web service

To rename a DynURL configuration file with the RESTful web service, issue an HTTP PUT command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to rename.
new_name	The new DynURL configuration file name.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/dynurl_config/{id}

PUT_DATA: {
"new_name": "new_dynurl_config_file_name"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Deleting a DynURL configuration file with web service

To delete a DynURL configuration file with the RESTful web service, issue an HTTP DELETE command on the DynURL Configuration File resource URI.

URL

```
https://{appliance_hostname}/dynurl_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to delete

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/dynurl_config/{id}
```

Response:

```
200 ok
```

Junction Mapping Table (JMT) configuration file management

Use either the local management interface or web service to manage Junction Mapping Table (JMT) configuration files.

Managing JMT configuration files with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.

Retrieving a list of all JMT configuration files with local management interface

To retrieve a list of all JMT configuration files with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. The file name, file size, and last modified time information of all JMT configuration files is displayed.

Viewing a JMT configuration file with local management interface

To view the contents of a JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. Select the file that you want to view.
3. Click **Edit**. The contents of the file is displayed in the pop-up window.

Creating a JMT configuration file with local management interface

To create a JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. Click **New**.
3. In the window that pops up:
   a. Modify the content of the file.
   b. Enter the name for the file to be created in the **JMT Configuration File Name** field.
   c. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing a new JMT configuration file with local management interface

To import a new JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.

2. Click **Manage** > **Import**.
3. In the window that pops up, click **Browse**.
4. Choose the file to import from your local workstation.
5. Click **Import**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting a JMT configuration file with local management interface

To export a JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. Select the JMT configuration file to export.
3. Click **Manage** > **Export**.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. In the window that pops up, confirm to save the file to your local workstation.

Updating an existing JMT configuration file with local management interface

To update an existing JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. Select the JMT configuration file to edit.
3. Click **Edit**.
4. In the window that pops up:
   a. Modify the file content.
   b. Click **Save**.

      Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Renaming a JMT configuration file with local management interface

To rename a JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. Select the JMT configuration file to rename.
3. Click **Manage** > **Rename**.

4. In the window that pops up, enter the new name for the file in the **New Resource Name** field.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting a JMT configuration file with local management interface

To delete a JMT configuration file with the local management interface, use the Junction Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Junction Mapping**.
2. Select the JMT configuration file to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing JMT configuration files with web service

Use the JMT Configuration File resource URI to manage JMT configuration files.

Retrieving a list of all JMT configuration files with web service

To retrieve a list of all JMT configuration files with the RESTful web service, issue an HTTP GET command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the JMT configuration file.
version	The current version of file. This information is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Response

HTTP response code and JSON data that represents the name and version information of the files.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/jmt_config
```

Response:

```
200 ok

[
{
"id":"jmt_config_file_1",
"version":"version_id"
},
{
"id":"jmt_config_file_2",
"version":"version_id"
},
...
]
```

Retrieving a JMT configuration file with web service

To retrieve a JMT configuration file with the RESTful web service, issue an HTTP GET command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to retrieve

Response

HTTP response code and JSON data that represents the file content.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/jmt_config/{id}
```

Response:

```
200 ok

{"contents":"File contents"}
```

Retrieving the JMT configuration file template with web service

To retrieve the JMT configuration file template with the RESTful web service, issue an HTTP GET command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/wga_templates/jmt_template
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the template data.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/jmt_template
```

Response:

```
200 ok

{"id": "TEMPLATE DATA"}]}
```

Creating a JMT configuration file with web service

To create a JMT configuration file with the RESTful web service, issue an HTTP POST command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
name	The new JMT Configuration file name. This name must be unique.
jmt_config_data	The data that is saved into the new JMT configuration file.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/jmt_config

POST_DATA: {
"name": new file name
"jmt_config_data": "<DATA>"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Importing a new JMT configuration file with web service

To import a new JMT configuration file with the RESTful web service, issue an HTTP POST command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
jmt_config_file	Name of the JMT configuration file to import.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/jmt_config

POST_DATA: {
"jmt_config_file": "JMT Configuration File to Import"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Exporting a JMT configuration file with web service

To export a JMT configuration file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to export

Response

HTTP response code and the file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/jmt_config/{id}?export
```

Response:

```
200 ok

The file text
```

Exporting the JMT configuration file template with web service

To export the JMT configuration file template with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/wga_templates/jmt_template?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the template file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/jmt_template?export
```

Response:

```
200 ok

The template file text
```

Updating an existing JMT configuration file with web service

To update an existing JMT configuration file with the RESTful web service, issue an HTTP PUT command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The existing JMT configuration file name.
jmt_config_data	The data that is saved into the existing JMT configuration file.
jmt_config_file	The file that contains the data to be saved into the existing JMT configuration file.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Pass the new file contents as JSON

```
PUT https://{appliance_hostname}/jmt_config/{id}

PUT_DATA: {
"jmt_config_data": "<DATA>"
}
```

Pass the new file contents as a file

```
PUT https://{appliance_hostname}/jmt_config/{id}

PUT_DATA: {
"jmt_config_file": The file containing the new
data as application/octet-stream
}
```

Response:

```
200 ok
```

Renaming a JMT configuration file with web service

To rename a JMT configuration file with the RESTful web service, issue an HTTP PUT command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	ID of the file to rename
new_name	The new JMT configuration file name.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/jmt_config/{id}

PUT_DATA: {
   "new_name": "new_jmt_config_file_name"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Deleting a JMT configuration file with web service

To delete a JMT configuration file with the RESTful web service, issue an HTTP DELETE command on the JMT Configuration File resource URI.

URL

```
https://{appliance_hostname}/jmt_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Parameter	Description
id	ID of the file to delete

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/jmt_config/{id}
```

Response:

```
200 ok
```

Client Certificate CDAS file management

Use either the local management interface or web service to manage Client Cert CDAS files.

Managing client certificate CDAS files with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.

Retrieving a list of all Client Certificate CDAS files with local management interface

To retrieve a list of all Client Certificate CDAS files with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. The file name, file size, and last modified time information of all Client Certificate CDAS files is displayed.

Creating a Client Certificate CDAS file with local management interface

To create a Client Certificate CDAS file with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. Click **New**.
3. In the window that pops up, enter the name for the file to be created in the **New Resource Name** field.
4. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing a Client Certificate CDAS file with local management interface

To import an existing Client Certificate CDAS file with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. Click **Manage** > **Import**.
3. In the window that pops up, click **Browse**.
4. Choose the file to import from your local workstation.
5. Click **Import**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting a Client Certificate CDAS file with local management interface

To export a Client Certificate CDAS file with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. Select the Client Certificate CDAS file to export.
3. Click **Manage** > **Export**.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. In the window that pops up, confirm to save the file to your local workstation.

Editing a Client Certificate CDAS file with local management interface

To edit a Client Certificate CDAS file with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. Select the Client Certificate CDAS file to edit.
3. Click **Edit**.
4. In the window that pops up:
   a. Modify the file content.
   b. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Renaming a Client Certificate CDAS file with local management interface

To rename a Client Certificate CDAS file with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. Select the Client Certificate CDAS file to rename.
3. Click **Manage** > **Rename**.
4. In the window that pops up, enter the new name for the file in the **New Resource Name** field.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting a Client Certificate CDAS file with local management interface

To delete a Client Certificate CDAS file with the local management interface, use the Client Certificate Mapping management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Client Certificate Mapping**.
2. Select the Client Certificate CDAS file to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing client certificate CDAS files with web service

Use the Client Cert CDAS file resource URI to manage client certificate CDAS files.

Retrieving a list of all Client Cert CDAS files with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the Client Cert CDAS files resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the name and version information of the files.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/client_cert_cdas
```

Response:

```
200 ok

[
{
"id":"client_cert_cdas_file_1",
"version":"version_id"
},
{
"id":" client_cert_cdas_file_2",
"version":"version_id"
},
...
]
```

Retrieving a Client Cert CDAS file with web service

To complete this operation with the RESTful web service, issue an HTTP GET command with on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the Client Cert CDAS file to retrieve.

Response

HTTP response code and JSON data that represents the content of the file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/client_cert_cdas/{id}
```

Response:

```
200 ok

{"contents":"file contents"}
```

Retrieving the Client Cert CDAS file template with web service

To retrieve the Client Cert CDAS file template with the RESTful web service, issue an HTTP GET command on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/wga_templates/
client_cert_cdas_template
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the template data.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/
client_cert_cdas_template
```

Response:

```
200 ok

{"id": "TEMPLATE DATA"}]}
```

Creating a Client Cert CDAS file with web service

To create a Client Cert CDAS file with the RESTful web service, issue an HTTP POST command on the Client Cert CDAS file resource URI. You can complete this operation by using a template or plain text.

URL

```
https://{appliance_hostname}/client_cert_cdas
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
name	The name of the new Client Cert CDAS file.
content	The content of the new Client Cert CDAS file as plain text. This parameter is optional. If not specified, the new file is created from the template file.

Response

HTTP response code and JSON data that represents the name of the created file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Using a template:

```
POST https://{appliance_hostname}/client_cert_cdas

POST_DATA {"name":"new_file_name"}
```

Using plain text:

```
POST https://{appliance_hostname}/client_cert_cdas

POST_DATA: {
"name":"<client_cert_cdas_file_name>",
"content":"<client_cert_cdas_file_content>"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Updating an existing Client Cert CDAS file with web service

To update an existing Client Cert CDAS file with the RESTful web service, issue an HTTP PUT command on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the Client Cert CDAS file to update.
content	The content of the new Client Cert CDAS file as plain text.
file	The Client Cert CDAS file as application or octet-stream.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Pass the new file contents as JSON:

```
PUT https://{appliance_hostname}/client_cert_cdas/{id}

PUT_DATA: {"content":"client_cert_cdas_file_content"}
```

Pass the new file contents as a file

```
PUT https://{appliance_hostname}/client_cert_cdas/{id}

PUT_DATA: { "file":"client_cert_cdas_file" }
```

Response:

```
200 ok
```

Importing a Client Cert CDAS file with web service

To import a Client Cert CDAS file with the RESTful web service, issue an HTTP POST command on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
file	The Client Cert CDAS file as application or octet-stream. It contains both the file name and data content of the Client Cert CDAS file to be imported.

Response

HTTP response code and JSON data that represents the imported file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/client_cert_cdas

POST_DATA: { "file":"client_cert_cdas_file" }
```

Response:

```
200 ok

{"id":"new file name"}
```

Exporting a Client Cert CDAS file with web service

To export a Client Cert CDAS file with the RESTful web service, issue an HTTP GET command with the `export` parameter on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas/{id}?export
```

Note: The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, the content of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the Client Cert CDAS file to export.

Response

HTTP response code and the exported file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/client_cert_cdas/{id}?export
```

Response:

```
200 ok

The file text
```

Exporting the Client Cert CDAS file template with web service

To export the Client Cert CDAS file template with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/wga_templates/
client_cert_cdas_template?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and template file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/
client_cert_cdas_template?export
```

Response:

```
200 ok

The template file text
```

Renaming a Client Cert CDAS file with web service

To rename a Client Cert CDAS file with the RESTful web service, issue an HTTP PUT command on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the Client Cert CDAS file to rename.
new_name	The new name for the Client Cert CDAS file.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/client_cert_cdas/{id}

PUT_DATA: { "new_name":" new_file_name" }
```

Response:

```
200 ok

{"id":"new file name"}
```

Deleting a Client Cert CDAS file with web service

To delete a Client Cert CDAS file with the RESTful web service, issue an HTTP DELETE command on the Client Cert CDAS file resource URI.

URL

```
https://{appliance_hostname}/client_cert_cdas/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the Client Cert CDAS file to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/client_cert_cdas/{id}
```

Response:

```
200 ok
```

Forms Based SSO (FSSO) configuration file management

Use either the local management interface or web service to manage Forms Based SSO (FSSO) configuration files.

Managing FSSO configuration files with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.

Retrieving a list of all FSSO configuration files with local management interface

To retrieve a list of all FSSO configuration files with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. The file name, file size, and last modified time information of all FSSO configuration files is displayed.

Viewing an FSSO configuration file with local management interface

To view the contents of an FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Select the file that you want to view.
3. Click **Edit**. The contents of the file is displayed in the pop-up window.

Creating an FSSO configuration file with local management interface

To create an FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Click **New**.
3. In the window that pops up:
   a. Modify the content of the file.
   b. Enter the name for the file to be created in the **FSSO Configuration File Name** field.
   c. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing a new FSSO configuration file with local management interface

To import a new FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Click **Manage** > **Import**.
3. In the window that pops up, click **Browse**.
4. Choose the file to import from your local workstation.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting an FSSO configuration file with local management interface

To export an FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Select the FSSO configuration file to export.
3. Click **Manage** > **Export**.

   Note: You must configure the software to block pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. In the window that pops up, confirm to save the file to your local workstation.

Updating an existing FSSO configuration file with local management interface

To update an existing FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Select the FSSO configuration file to edit.
3. Click **Edit**.
4. In the window that pops up:
   a. Modify the file content.
   b. Click **Save**.

      Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Renaming an FSSO configuration file with local management interface

To rename an FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Select the FSSO configuration file to rename.
3. Click **Manage** > **Rename**.
4. In the window that pops up, enter the new name for the file in the **New Resource Name** field.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting an FSSO configuration file with local management interface

To delete an FSSO configuration file with the local management interface, use the Forms Based Single Sign-on management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **Forms Based Single Sign-on**.
2. Select the FSSO configuration file to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing FSSO configuration files with web service

Use the FSSO Configuration File resource URI to manage FSSO configuration files.

Retrieving a list of all FSSO configuration file with web service

To retrieve a list of all FSSO configuration file with the RESTful web service, issue an HTTP GET command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the name and version information of the files.

```
[
{
"id":"fsso_config_file_1",
"version":"version_id"
},
{
"id":"fsso_config_file_2",
"version":"version_id"
},
...
]
```

Parameter	Description
id	Name of the FSSO configuration file.
version	The current version of file. This information is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/fsso_config
```

Response:

```
200 ok

[
{
"id":"fsso_config_file_1",
"version":"version_id"
},
{
"id":"fsso_config_file_2",
"version":"version_id"
},
...
]
```

Retrieving an FSSO configuration file with web service

To retrieve an FSSO configuration file with the RESTful web service, issue an HTTP GET command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the FSSO configuration file to retrieve.

Response

HTTP response code and JSON data that represents the file content.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/fsso_config/{id}
```

Response:

```
200 ok

{"fsso_config_data": "DATA"}
```

Retrieving the FSSO configuration file template with web service

To retrieve the FSSO configuration file template with the RESTful web service, issue an HTTP GET command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/wga_templates/fsso_template
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents template file data.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/fsso_template
```

Response:

```
200 ok

{"id": "TEMPLATE DATA"}]}
```

Creating an FSSO configuration file with web service

To create an FSSO configuration file with the RESTful web service, issue an HTTP POST command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
name	The new FSSO Configuration file name. This name must be unique.
fsso_config_data	The data that is saved into the new FSSO configuration file.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/fsso_config

POST_DATA: {
"name": "new file name"
"fsso_config_data": "<DATA>"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Importing a new FSSO configuration file with web service

To import a new FSSO configuration file with the RESTful web service, issue an HTTP POST command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
fsso_config_file	Name of the file to import.

Response

HTTP response code and JSON data that represents the name of the imported file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/fsso_config

POST_DATA: {
"fsso_config_file": "FSSO Configuration File to Import"
(as application/octet-stream)
}
```

Response:

```
200 ok

{"id":"new file naem"}
```

Exporting an FSSO configuration file with web service

To export an FSSO configuration file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the FSSO configuration file to export.

Response

HTTP response code and the file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/fsso_config/{id}?export
```

Response:

```
200 ok

The file text
```

Exporting the FSSO configuration file template with web service

To export the FSSO configuration file template with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/wga_templates/fsso_template?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the template file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/fsso_template?export
```

Response:

```
200 ok

The file text
```

Updating an existing FSSO configuration file with web service

To update an existing FSSO configuration file with the RESTful web service, issue an HTTP PUT command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The existing FSSO configuration file name.
fsso_config_data	The data that is saved into the existing FSSO configuration file.
fsso_config_file	The file that contains the data to be saved into the existing FSSO configuration file.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Pass the new file contents as JSON

```
PUT https://{appliance_hostname}/fsso_config/{id}

PUT_DATA: {
"fsso_config_data": "DATA"
}
```

Pass the new file contents as a file

```
PUT https://{appliance_hostname}/fsso_config/{id}

PUT_DATA: {
"fsso_config_file": The file containing the new
data as application/octet-stream
}
```

Response:

```
200 ok
```

Renaming an FSSO configuration file with web service

To rename an FSSO configuration file with the RESTful web service, issue an HTTP PUT command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the file to rename.
new_name	The new FSSO configuration file name.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/fsso_config/{id}

PUT_DATA: {
   "new_name": "new_fsso_config_file_name"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Deleting an FSSO configuration file with web service

To delete an FSSO configuration file with the RESTful web service, issue an HTTP DELETE command on the FSSO Configuration File resource URI.

URL

```
https://{appliance_hostname}/fsso_config/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the file to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/fsso_config/{id}
```

Response:

```
200 ok
```

HTTP Transformation Rule file management

Use either the local management interface or web service to manage HTTP Transformation Rule files.

Managing HTTP transformation rule files with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.

Retrieving a list of all HTTP Transformation Rule files with local management interface

To retrieve a list of all HTTP Transformation Rule files with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.
2. The file name, file size, and last modified time information of all HTTP Transformation Rule files is displayed.

Creating an HTTP Transformation Rule file with local management interface

To create an HTTP Transformation Rule file with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.

2. Click **New**.
3. In the window that pops up, enter the name for the file to be created in the **New Resource Name** field.
4. Select to use either **Request** or **Response** as the template.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing an HTTP Transformation Rule file with local management interface

To import an existing HTTP Transformation Rule file with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.
2. Click **Manage** > **Import**.
3. In the window that pops up, click **Browse**.
4. Choose the file to import from your local workstation.
5. Click **Import**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting an HTTP Transformation Rule file with local management interface

To export an HTTP Transformation Rule file with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.
2. Select the HTTP Transformation Rule file to export.
3. Click **Manage** > **Export**.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. In the window that pops up, confirm to save the file to your local workstation.

Editing an HTTP Transformation Rule file with local management interface

To edit an HTTP Transformation Rule file with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.
2. Select the HTTP Transformation Rule file to edit.
3. Click **Edit**.
4. In the window that pops up:

a. Modify the file content.
b. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Renaming an HTTP Transformation Rule file with local management interface

To rename an HTTP Transformation Rule file with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.
2. Select the HTTP Transformation Rule file to rename.
3. Click **Manage** > **Rename**.
4. In the window that pops up, enter the new name for the file in the **New Resource Name** field.
5. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting an HTTP Transformation Rule file with local management interface

To delete an HTTP Transformation Rule file with the local management interface, use the HTTP Transformation management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Settings** > **HTTP Transformation**.
2. Select the HTTP Transformation Rule file to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing HTTP transformation rule files with web service

Use the HTTP Transformation Rule file resource URI to manage HTTP transformation settings.

Retrieving a list of all HTTP Transformation Rule files with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the HTTP Transformation Rule files resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

GET

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the name and version information of the files.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/http_transformation_rules
```

Response:

```
200 ok

[
{
"id":"http_transformation_rule_file_1",
"version":"version_id"
},
{
"id":"http_transformation_rule_file_2",
"version":"version_id"
},
...
]
```

Retrieving an HTTP Transformation Rule file template with web service

To retrieve the HTTP Transformation Rule file template with the RESTful web service, issue an HTTP GET command on the HTTP Transformation Rule file resource URI.

URL

Retrieve the request template

```
https://{appliance_hostname}/wga_templates/
{http_transformation_rules_request_template}
```

Retrieve the response template

```
https://{appliance_hostname}/wga_templates/
{http_transformation_rules_response_template}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the name and version information of the files.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Retrieve the request template

```
GET https://{appliance_hostname}/wga_templates/
http_transformation_rules_request_template
```

Retrieve the response template

```
GET https://{appliance_hostname}/wga_templates/
http_transformation_response_request_template
```

Response:

```
200 ok

{"id":"template data"}
```

Retrieving an HTTP Transformation Rule file with web service

To retrieve an HTTP Transformation Rule file with the RESTful web service, issue an HTTP GET command on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the file to export.

Response

HTTP response code and JSON data that represents the file content.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/http_transformation_rules/{id}
```

Response:

```
200 ok

{"http_transformation_rules_data": "DATA"}
```

Creating an HTTP Transformation Rule file with web service

To create an HTTP Transformation Rule file with the RESTful web service, issue an HTTP POST command on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
name	The name of the new HTTP Transformation Rule file. This name must be unique.
template	The template to create the file from. Valid values are "request" and "response".
content	This is the content of the new HTTP Transformation Rule file as plain text.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Using a template

```
POST https://{appliance_hostname}/http_transformation_rules

POST_DATA {
"name":"new_file_name",
"template":"template_name"
}
```

Using plain text

```
POST https://{appliance_hostname}/http_transformation_rules

POST_DATA: {
"name":"http_transformation_rules_file_name",
"content":"http_transformation_rules_file_content"
}
```

Response:

```
200 ok

{"id":"new file name"}
```

Importing an HTTP Transformation Rule file with web service

To import an HTTP Transformation Rule file with the RESTful web service, issue an HTTP POST command on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
file	The HTTP Transformation Rule file as application or octet-stream. It contains both the file name and data content of the HTTP Transformation Rule file to be imported.

Response

HTTP response code and JSON data that represents the imported file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/http_transformation_rules

POST_DATA: { "file":"http_transformation_rules_file" }
```

Response:

```
200 ok

{"id":"new file name"}
```

Exporting an HTTP Transformation Rule file with web service

To export an HTTP Transformation Rule file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, the content of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the file to export.

Response

HTTP response code and the file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/http_transformation_rules
/{id}?export
```

Response:

```
200 ok

The file text
```

Exporting an HTTP Transformation Rule file template with web service

To export the HTTP Transformation Rule file template with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the HTTP Transformation Rule file resource URI.

URL

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, a snippet of the file is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the template file text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Retrieve the request template

```
GET https://{appliance_hostname}/wga_templates/
http_transformation_rules_request_template?export
```

Retrieve the response template

```
GET https://{appliance_hostname}/wga_templates/
http_transformation_response_request_template?export
```

Response:

```
200 ok

The template file text
```

Updating an existing HTTP Transformation Rule file with web service

To edit an existing HTTP Transformation Rule file with the RESTful web service, issue an HTTP PUT command on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the HTTP Transformation Rule file to update.
content	The content of the new HTTP Transformation Rule file as plain text.
file	The HTTP Transformation Rule file as application or octet-stream.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Pass the new file contents as JSON

```
PUT https://{appliance_hostname}/http_transformation_rules/{id}

PUT_DATA: {"content":"http_transformation_rule_file_content"}
```

Pass the new file contents as a file

```
PUT https://{appliance_hostname}/http_transformation_rules/{id}

PUT_DATA: { "file":"http_transformation_rule_file" }
```

Response:

```
200 ok
```

Renaming an HTTP Transformation Rule file with web service

To rename an HTTP Transformation Rule file with the RESTful web service, issue an HTTP PUT command on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the HTTP Transformation Rule file to rename.
new_name	The new name for the HTTP Transformation Rule file.

Response

HTTP response code and JSON data that represents the new file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/http_transformation_rules/{id}

PUT_DATA: { "new_name":" new_file_name" }
```

Response:

```
200 ok

{"id":"new file name"}
```

Deleting an HTTP Transformation Rule file with web service

To delete an HTTP Transformation Rule file with the RESTful web service, issue an HTTP DELETE command on the HTTP Transformation Rule file resource URI.

URL

```
https://{appliance_hostname}/http_transformation_rules/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Parameter	Description
id	Name of the file to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/http_transformation_rules/{id}
```

Response:

```
200 ok
```

SSL certificates

Use either the local management interface or web service to manage SSL certificates.

Managing SSL certificates with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.

Listing current certificate database names with local management interface

To list all current certificate database names with the local management interface, use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. You can view all current certificate database names and their last modified time information.

Adding description to a certificate database with local management interface

To add a description to a certificate database with the local management interface, use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database that you want to describe.
3. Select **Manage** > **Describe**.
4. In the Describe SSL Certificates Database window, enter the description of the certificate database.
5. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Creating a certificate database with local management interface

To create a certificate database with the local management interface, use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. From the menu bar, click **New**.
3. On the Create SSL Certificate Database page, enter the name of the certificate database that you want to create. The name of the certificate database name must be unique.
4. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing a certificate database with local management interface

To import a certificate database with the local management interface, use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select **Manage** > **Import**.
3. Click **Browse** under **Certificate Database File**.
4. Browse to the directory that contains the file to be imported and select the file. Click **Open**.
5. Click **Browse** under **Stash File**.
6. Browse to the directory that contains the file to be imported and select the file. Click **Open**.
7. Click **Import**. A message that indicates successful import is displayed.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting a certificate database with local management interface

To export a certificate database with the local management interface, use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database that you want to export.
3. Select **Manage** > **Export**.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. Confirm the save operation when the browser prompts you to save the .zip file.

Renaming a certificate database with local management interface

To rename a certificate database with the local management interface, use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database that you want to rename.
3. Select **Manage** > **Rename**
4. In the Rename SSL Certificates Database window, enter the new name of the certificate database. The new name of the certificate database name must be unique.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting a certificate database with local management interface

To delete a certificate database with the local management interface, you can use the SSL Certificates management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database that you want to delete.
3. Select **Delete**
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing signer certificates in a certificate database with local management interface

To manage signer certificates in a certificate database, you can use the SSL Certificates management page. In particular, you can import, export, or delete signer certificates, and list all signer certificate names.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database of interest.
3. Select **Manage** > **Edit SSL Certificate Database**.
4. All signer certificate names are displayed on the **Signer Certificates** tab.

   Import a signer certificate

   a. Click **Manage** > **Import**.
   b. Click **Browse**. Then, select the signer certificate to be imported.
   c. In the **Certificate Label** field, enter what you want to label the signer certificate.
   d. Select **Trusted** if you want to set the signer certificate as trusted.
   e. Click **Import**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

View and export a signer certificate

a. Select the signer certificate that you want to view.
b. Click **Manage** > **View**. The content of the signer certificate is displayed in the browser.
c. *Optional:* Click **Export**. Then, confirm the save operation in the window that pops up.

Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up widnows for the appliance before files can be exported.

Export a signer certificate

a. Select the signer certificate that you want to export.
b. Click **Manage** > **Export**.
c. Confirm the save operation in the browser window that pops up.

Delete a signer certificate

a. Select the signer certificate that you want to delete.
b. Click **Delete**.
c. In the window that pops up, click **Yes**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Set a signer certificate as trusted or not trusted

a. Select the signer certificate that you want to edit.
b. Click **Edit**.
c. Select or clear the **Trusted** check box to set the signer certificate as trusted or not trusted.
d. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing personal certificates in a certificate database with local management interface

To manage personal certificates in a certificate database with the local management interface, use the SSL Certificates management page. In particular, you can import, view, export, or delete personal certificates, list all personal certificate names, and create self-signed personal certificates.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database of interest.
3. Select **Manage** > **Edit SSL Certificate Database**.
4. Click the **Personal Certificates** tab. All personal certificate names are displayed on this tab.

Import a personal certificate

a. Click **Manage** > **Import**.

b. Click **Browse**. Then, select the file that contains the personal certificate to import.
c. *Optional:* Specify the password for the file that contains the personal certificate to import.
d. Click **Import**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Receive a personal certificate

Note: A personal certificate can be received only if a corresponding certificate request exists.

a. Click **Manage** > **Recieve**.
b. Click **Browse**. Then, select the personal certificate to be received.
c. Select the **Default** check box if you want to set the personal certificate as default.
d. Click **Receive**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

View a personal certificate

a. Select the personal certificate you want to view.
b. Click **Manage** > **View**. The content of the personal certificate is displayed in the browser.
c. *Optional:* Click **Export**. Then, confirm the save operation in the window that pops up.

Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.

Export a personal certificate

a. Select the personal certificate that you want to export.
b. Click **Manage** > **Export**.

Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.

c. Confirm the save operation in the browser window that pops up.

Delete a personal certificate

a. Select the personal certificate that you want to delete.
b. Click **Delete**.
c. In the window that pops up, click **Yes**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Create a personal certificate (self-signed)

a. Click **New**.

b. Enter **Certificate Label**, **Certificate Distinguished Name**, **Key Size**, and **Expiration Time**. The default value for **Expiration Time** is 365 days.
c. Select the **Default** check box if you want to set this personal certificate as the default certificate.
d. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Set a personal certificate as default

a. Select the personal certificate that you want to edit.
b. Click **Edit**.
c. Select the **Set as the Default Certificate** check box to set the personal certificate as the default certificate.
d. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing certificate requests in a certificate database with local management interface

To manage certificate requests in a certificate database with the local management interface, use the SSL Certificates management page. In particular, you can create, view, export, or delete certificate requests, and list all certificate request names.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSL Certificates**.
2. Select the certificate database of interest.
3. Select **Manage** > **Edit SSL Certificate Database**.
4. Click the **Certificate Requests** tab. All certificate request names are displayed on this tab.

Create a certificate request

a. Click **New**.
b. Enter **Certificate Request Label**, **Certificate Request Distinguished Name**, and **Key Size**.
c. Click **Save**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

View and export a certificate request

a. Select the certificate request that you want to view.
b. Click **Manage** > **View**. The content of the certificate request is displayed in the browser.
c. *Optional:* Click **Export**. Then, confirm the save operation in the window that pops up.

Export a certificate request

a. Select the certificate request that you want to export.
b. Click **Manage** > **Export**. The content of the certificate request is displayed in the browser.

c. Confirm the save operation in the window that pops up.

Delete a certificate request

a. Select the certificate request that you want to delete.
b. Click **Delete**.
c. In the window that pops up, click **Yes**.

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing SSL certificates with web service

Use the SSL Certificates resource URI to manage SSL certificate databases.

Retrieving all current certificate database names with web service

To retrieve the certificate database names with the RESTful web service, issue an HTTP GET command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the certificate database name, version, and description information.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates
```

Response:

```
200 ok

[
{
"id":"file_1",
"version":"version_id"
"description":"The certificate database description"
},
{
```

```
"id":"file_2",
"version":"version_id"
"description":"The certificate database description"
},
...
]
```

Parameter	Description
id	Name of the certificate database.
version	The current version of file. This information is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).
description	The description that is set for this certificate database.

Creating a certificate database with web service

To create a certificate database with the RESTful web service, issue an HTTP POST command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_name	The new certificate database name that is used to uniquely identify the certificate database. This parameter is required.

Response

HTTP response code and JSON data that represents the new certificate database name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ssl_certificates

POST_DATA: {
"kdb_name": "junctionkdb"
}
```

Response:

```
200 ok

{"id":"The new certificate database name"}
```

Importing a certificate database with web service

To import a certificate database with the RESTful web service, issue an HTTP POST command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb	The certificate database file to be imported. This parameter is required.
stash	The stash file to be imported. This parameter is required.

Response

HTTP response code and JSON data that represents the new certificate database name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ssl_certificates

POST_DATA: {
"kdb":"/example/cert.kdb" (as application/octet-stream),
"stash":"/example/cert.sth" (as application/octet-stream)
}
```

Response:

```
200 ok

{"id":"The new certificate database name"}
```

Exporting a certificate database with web service

To export a certificate database and stash file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{id}?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the certificate database to export.

Response

HTTP response code and a .zip file that contains the certificate database and stash file.

Notes:

- For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.
- Redirect the output of the web service to a local file as it is not screen readable.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{id}?export
```

Response:

```
200 ok

The zip file
```

Setting a description for a certificate database with web service

To add a description to a certificate database with the RESTful web service, issue an HTTP PUT command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the certificate database to set description for.
description	The description of the certificate database.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/ssl_certificates/{id}

PUT_DATA: {
"description": "Descripton of the certificate database"
}
```

Response:

```
200 ok
```

Renaming a certificate database with web service

To rename a certificate database with the RESTful web service, issue an HTTP PUT command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The name of the certificate database to rename.
new_name	The new name for the certificate database. This parameter is required.

Response

HTTP response code and JSON data that represents the new certificate database name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/ssl_certificates/{id}

PUT_DATA: {
"new_name": "newjunctionkdb"
}
```

Response:

```
200 ok

{"id":"the new certificate database name"}
```

Deleting a certificate database with web service

To rename a certificate database with the RESTful web service, issue an HTTP DELETE command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the certificate database to delete.

Response

HTTP response code and JSON data error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/ssl_certificates/{id}
```

Response:

```
200 ok
```

Retrieving signer certificate names and details in a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the SSL Certificates signer certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name the certificate database to retrieve information for.

Response

HTTP response code and JSON data that represents signer certificate details.

Parameter	Description
id	Name of the signer certificate.
trusted	Whether the certificate is trusted. Valid values are "true" and "false".
version	Version of the signer certificate.
issuer	Issuer of the signer certificate.
subject	Subject of the signer certificate.
notbefore	Start date of the validity of the signer certificate.
notafter	Expiry date of the validity of the signer certificate.
keysize	The key size of the signer certificate.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert
```

Response:

```
200 ok

[
{
```

```
"id": "cert_name",
"trusted": true/false,
"version":"cert_version",
"issuer":"cert_issuer",
"subject":"cert_subject",
"notbefore":"date",
"notafter":"date",
"keysize":"the key size of the certificate"
},
{
"id": "cert_name2",
"trusted": true/false,
"version":"cert_version",
"issuer":"cert_issuer",
"subject":"cert_subject",
"notbefore":"date",
"notafter":"date",
"keysize":"the key size of the certificate"
},
{
....
}
]
```

Retrieving a signer certificate from a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the SSL Certificates signer certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the signer certificate.

Response

HTTP response code and JSON data that represents the certificate content.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert/{id}
```

Response:

```
200 ok

{"contents":"The certificate text"}
```

Importing a signer certificate into a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP POST command on the SSL Certificates signer certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
label	The new signer certificate label. A name that identifies the signer certificate. There can be many signer certificates in a certificate database so this name must be unique. This parameter is required.
trust	Whether the trust field on the signer certificate is set. The value is "yes" or "no".
cert	The signer certificate to be imported. This value can be either a string or a file. This parameter is required.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert

POST_DATA: {
"label":"webseal.ibm.com",
"trust":"yes",
"cert":"The certificate to import" (as string or file)}
```

Response:

```
200 ok
```

Exporting a signer certificate from a certificate database with web service

To export a signer certificate from a certificate database with the RESTful web service, issue an HTTP GET command with the`export` parameter on the SSL Certificates signer certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert
/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the signer certificate is exported. If no `export` parameter is included in the URL, the content of the signer certificate is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the signer certificate.

Response

HTTP response code and the certificate text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert
/{id}?export
```

Response:

```
200 ok

The certificate text
```

Setting whether a signer certificate is trusted in a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP PUT command on the SSL Certificates signer certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the signer certificate.
trusted	Whether to set the signer certificate as trusted or not. The value is "yes" or "no". This parameter is required.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert/{id}

PUT_DATA: {
"trusted": "yes"
}
```

Response:

```
200 ok
```

Deleting a signer certificate from a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP DELETE command on the SSL Certificates signer certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the signer certificate.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/ssl_certificates/{kdb_id}/signer_cert/{id}
```

Response:

```
200 ok
```

Retrieving personal certificate names and details in a certificate database with web service

To complete the operation with the RESTful web service, issue an HTTP GET command on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.

Response

HTTP response code and JSON data that represents the details of personal certificates.

Parameter	Description
id	Name of the personal certificate.
default	Whether this personal certificate is default. Valid values are "true" or "false".

Parameter	Description
version	Version of the personal certificate.
issuer	Issuer of the personal certificate.
subject	Subject of the personal certificate.
notbefore	Start date of the personal certificate validity.
notafter	Expire date of the personal certificate validity.
keysize	Key size of the personal certificate.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert
```

Response:

```
200 ok

[
{
"id": "cert_name",
"default":true/false,
"version":"cert_version",
"issuer":"cert_issuer",
"suject":"cert_subject",
"notbefore":"date",
"notafter":"date",
"keysize":"the key size of the certificate"
},
{
"id": "cert_name2",
"default":true/false,
"version":"cert_version",
"issuer":"cert_issuer",
"suject":"cert_subject",
"notbefore":"date",
"notafter":"date".
"keysize":"the key size of the certificate"
},
{
....
}
]
```

Retrieving a personal certificate from a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

GET

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the personal certificate to retrieve.

Response

HTTP response code and JSON data that represents the content of the certificate.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}
```

Response:

```
200 ok

{"contents":"The certificate text"}
```

Receiving a personal certificate into a certificate database with web service

To complete the operation with the RESTful web service, issue an HTTP POST command on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

POST

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
operation	The type of create operation to perform. In this case, the value is "receive". This parameter is required.

Parameter	Description
default	Defines whether this new certificate is the default certificate. The value is “yes” or “no”.
cert	The personal certificate to be received. This value can be either a string or a file. This parameter is required.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

Passing a string

```
POST https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert

POST_DATA: {
"default":"yes",
"operation":"receive",
"cert":"-----BEGIN CERTIFICATE-----
MIIB7TCCAVagAwIBAgIID4yVO7xxFmUwDQYJKoZIhvcNAQEEBQAwOTELMAkGA1UE
BhMCVVMxGDAWBgNVBAoTD1BvbGljeSBEaXJlY3RvcjEQMA4GA1UEAxMHd2Vic2Vh
bDAeFw0xMjEwMjkyMzU2NThaFw0zMjEwMjUyMzU2NThaMDkxCzAJBgNVBAYTAlVT
MRgwFgYDVQQKEw9Qb2xpY3kgRGlyZWN0b3IxEDAOBgNVBAMTB3dlYnNlYWwwgZ8w
DQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANDboxdjSm+farEA3+ZIPcJxEbkp2RxX
uEM0Rbu3GUQxShFS8cFWI6LSP8WPCCRCl32eYuPjcEyFYxS/xVJZFsIwlk5IKmLv
uqOPUGr7XFSh/O+XJM+hqt/Y+hWn4XmhR6rqkubVSRlO+DS5WzFJexvTrgs6IJCX
qwC1Ydi+gvx1AgMBAAEwDQYJKoZIhvcNAQEEBQADgYEArTOrvuQm9CaKOwK2j5s4
NY2PSfRhVUWU2gBLf8nKzxt1xRkiWPUfeSfFshYMmHjRGA3EiFVPLCMPkvn0nHba
dUM29Wva+wgfDwpUIJYW8WwJuYpRnvW8xu9302L8Mwi/irBULqTIuK9ZUCnaDCvZ
+K1NZwE3/fQ92nG67+TFzaY=
-----END CERTIFICATE-----"
}
```

Passing a file

```
POST https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert

POST_DATA: {
"operation":"receive",
"default":"yes"
"cert":"The certificate to receive" (as application/octet-stream),
}
```

Response:

```
200 ok
```

Importing a personal certificate into a certificate database with web service

To complete the operation with the RESTful web service, issue an HTTP POST command on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Hostname of the appliance.
kdb_id	Name of the certificate database.
operation	The type of create operation to perform. In this case, the value is "import". This parameter is required.
cert	The personal certificate to be imported. This value can be either a string or a file. This parameter is required.
password	The password of the source certificate container

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert

POST_DATA: {
"operation":"import",
"cert":"/tmp/cert.pem" (as application/octet-stream),
"password":"The password of the source certificate container"
}
```

Response:

```
200 ok
```

Exporting a personal certificate from a certificate database with web service

To export or retrieve a personal certificate from a certificate database with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the personal certificate is exported. If no `export` parameter is included in the URL, the content of the personal certificate is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the personal certificate.

Response

HTTP response code and the certificate text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}?export
```

Response:

```
200 ok

The certificate text
```

Setting a personal certificate as default in a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP PUT command on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the personal certificate.
default	An indication that this certificate is set as the default certificate. The only supported value is "yes".

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}

PUT_DATA {
"default":"yes"
}
```

Response:

```
200 ok
```

Deleting a personal certificate from a certificate database with web service

To delete a personal certificate from a certificate database with the RESTful web service, issue an HTTP DELETE command on the SSL Certificates resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the personal certificate to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert/{id}
```

Response:

```
200 ok
```

Generating a self-signed personal certificate in a certificate database with web service

To complete this operation with RESTful web service, issue an HTTP POST command on the SSL Certificates personal certificate resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in “Configuration changes commit process” on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
operation	The type of create operation to perform. In this case, the value is "generate". This parameter is required.
label	The new personal certificate label that is used to uniquely identify the personal certificate. This parameter is required.
dn	Distinguished name. This parameter is required.
expire	The validity period, in days, for the new certificate. This parameter is required.
default	Whether the generated certificate is the default. The value is “yes” or “no”.
size	The size of the new key pair to be created. Valid values are 512, 1024, 2048 or 4096. This parameter is optional.

Response

HTTP response code and JSON data that represents the new certificate label.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ssl_certificates/{kdb_id}/personal_cert

POST_DATA: {
"operation":"generate",
"label": "cert.ibm.com",
"dn":"cn=testuser,o=ibm,c=us",
"expire":"365"
"default":"Yes"
"size":"The key size"
}
```

Response:

```
200 ok

{"id":"The new certificate label"}
```

Retrieving certificate request names and details in a certificate database with web service

To retrieve certificate requests from a certificate database with the RESTful web service, issue an HTTP GET command on the SSL Certificates certificate request resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the certificate request.
subject	Subject of the certificate request.
keysize	Key size of the certificate request.

Response

HTTP response code and JSON data that represents the certificate request details.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request
```

Response:

```
200 ok

[
{
"id": "request_name",
"subject":"request_subject",
"keysize":"the key size of the certificate"
},
{
"id": "request_name2",
"subject":"request_subject".
"keysize":"the key size of the certificate"
},
{
....
}
]
```

Retrieving a certificate request from a certificate database with the web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the SSL Certificates certificate request resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the certificate request to export.

Response

HTTP response code and JSON data that represents the certificate text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request/{id}
```

Response:

```
200 ok

{"contents":"The certificate request text"}
```

Creating a certificate request in a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP POST command on the SSL Certificates certificate request resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
label	The new certificate request label that is used to uniquely identify the personal certificate.
dn	Distinguished name.
size	The size of the new key pair to be created. Valid values are 512, 1024, 2048 or 4096. This parameter is optional.

Response

HTTP response code and JSON data that represents the new certificate request label.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request

POST_DATA: {
"label": "request.ibm.com",
"dn":"cn=testuser,o=ibm,c=us"
"size":"The key size"
}
```

Response:

```
200 ok

{"id":"The new certificate request label"}
```

Exporting a certificate request from a certificate database with the web service

To export a certificate request from a certificate database with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the SSL Certificates certificate request resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. The difference is that the export response is the text of the certificate request, while the retrieve response is the certificate request text in JSON format.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the certificate request to export.

Response

HTTP response code and the certificate text.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request/{id}?export
```

Response:

```
200 ok

The certificate text
```

Deleting a certificate request from a certificate database with web service

To complete this operation with the RESTful web service, issue an HTTP DELETE command on the SSL Certificates certificate request resource URI.

URL

```
https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
kdb_id	Name of the certificate database.
id	Name of the certificate request to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/ssl_certificates/{kdb_id}/cert_request/{id}
```

Response:

```
200 ok
```

SSO key management

Use either the local management interface or web service to manage SSO key files.

Managing SSO keys with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Keys** > **SSO Keys**.

Listing current SSO key files with local management interface

To list all current SSO key files with the local management interface, use the SSO Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSO Keys**.
2. You can view all current SSO key files and their last modified time information.

Creating an SSO key file with local management interface

To create an SSO key file with the local management interface, use the SSO Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSO Keys**.
2. Click **New**.
3. Enter the name of the key database that you want to create and then click **Save**. The name of the key database must be unique.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Importing an SSO key file with local management interface

To import an existing SSO key file, use the SSO Key Files management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSO Keys**.
2. Select **Manage** > **Import**.
3. Click **Browse**.
4. Browse to the directory that contains the file to be imported and select the file.
5. Click **Import**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting an SSO key file with local management interface

To export an SSO key file with the local management interface, use the SSO Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSO Keys**.
2. Select the SSO Key File that you want to delete.
3. Select **Manage** > **Export**. You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. Confirm the save operation if your browser displays a confirmation window.

Deleting an SSO key file with local management interface

To delete an existing SSO key file with the local management interface, use the SSO Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **SSO Keys**.
2. Select the file that you want to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing SSO keys with web service

Use the SSO Keys resource URI to manage SSO keys.

Listing all current SSO key files with web service

To retrieve the SSO key file names, issue an HTTP GET command on the SSO Keys resource URI.

URL

```
https://{appliance_hostname}/sso_key
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON date representing the name and version information of the SSO key files.

Parameter	Description
id	Name of the SSO key file.
version	The current version of the file. This is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/sso_key
```

Response:

```
200 ok

[
{
"id":"sso_key_file_1",
"version":"version_id"
},
{
"id":"sso_key_file_2",
"version":"version_id"
},
...
]
```

Creating an SSO key file with web service

To create an SSO key file with the RESTful web service, issue an HTTP POST command on the SSO Keys resource URI.

URL

```
https://{appliance_hostname}/sso_key
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
ssokey_name	The name that is used to uniquely identify the SSO Key File.

Response

HTTP response code and JSON date that represents the new SSO key file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/sso_key

POST_DATA: {
"ssokey_name": "ssokey_name"
}
```

Response:

```
200 ok

{"id":"The new SSO key file name"}
```

Importing an SSO key file with web service

To import an SSO key file with web service, issue an HTTP POST command on the SSO Keys resource URI.

URL

```
https://{appliance_hostname}/sso_key
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
sso_keyfile	The SSO key file to import.

Response

HTTP response code and JSON date that represents the imported SSO key file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/sso_key

POST_DATA: {
"sso_keyfile": "SSO Key File"}
```

Response:

```
200 ok

{"id":"The imported SSO key file name"}
```

Exporting an SSO key file with web service

To export an SSO key file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the SSO Keys resource URI.

URL

```
https://{appliance_hostname}/sso_key/{id}?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the SSO key file to export.

Response

HTTP response code and the SSO key file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/sso_key/{id}?export
```

Response:

```
200 ok

The SSO key file
```

Deleting an SSO key file with web service

To delete an SSO key file with the RESTful web service, issue an HTTP DELETE command on the SSO Keys resource URI.

URL

```
https://{appliance_hostname}/sso_key/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the SSO key file to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/sso_key/{id}
```

Response:

```
200 ok
```

LTPA keys

Use either the local management interface or web service to manage LTPA key files.

Managing LTPA keys with local management interface

In the local management interface, go to **Secure Reverse Proxy Settings** > **Global Keys** > **LTPA Keys**.

Retrieving all current LTPA key files with local management interface

To retrieve all current LTPA key files with the local management interface, use the LTPA Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **LTPA Keys**.
2. All current LTPA key files and their last modified time information are displayed.

Importing an LTPA key file with local management interface

To import an LTPA key file with the local management interface, use the LTPA Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **LTPA Keys**.
2. Click **Manage** > **Import**.
3. In the window that pops up, click **Browse**.
4. Select the file that you want to import and then click **Import**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Exporting an LTPA key file with local management interface

To export an LTPA key file with the local management interface, use the LTPA Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **LTPA Keys**.
2. Select the LTPA key file to export.
3. Click **Manage** > **Export**.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. Confirm the save operation in when the browser prompts you to save the file.

Renaming an LTPA key file with local management interface

To rename an LTPA key file with the local management interface, use the LTPA Keys management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **LTPA Keys**.
2. Select the LTPA key file to rename.
3. Click **Manage** > **Rename**.

4. In the window that pops up, enter the new name in the **New Resource Name** field.
5. Click **Save**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Deleting an LTPA key file with local management interface

To delete an LTPA key file with the local management interface, use the LTPA Key Files management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Global Keys** > **LTPA Keys**.
2. Select the LTPA key file to delete.
3. Click **Delete**.
4. In the window that pops up, click **Yes**.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Managing LTPA keys with web service

Use the LTPA Key resource URI to manage LTPA key files.

Retrieving all current LTPA key files with web service

To retrieve all LTPA key files with the RESTful web service, issue an HTTP GET command on the LTPA Keys resource URI.

URL

```
https://{appliance_hostname}/ltpa_key
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents name and version information of the LTPA key files.

Parameter	Description
id	Name of the LTPA key file.
version	The current version of file. This information is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ltpa_key
```

Response:

```
200 ok

[
{
"id":"ltpa_key_file_1",
"version":"version_id"
},
{
"id":"ltpa_key_file_2",
"version":"version_id"
},
...
]
```

Importing an LTPA key file with web service

To import an LTPA key file with the RESTful web service, issue an HTTP POST command on the LTPA Keys resource URI.

URL

```
https://{appliance_hostname}/ltpa_key
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
ltpa_keyfile	The LTPA key file to import.

Response

HTTP response code and JSON data that represents the new LTPA key file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/ltpa_key

POST_DATA: {
"ltpa_keyfile": "LTPA Key File to Import"
}
```

Response:

```
200 ok

{"id":"The new LTPA key file name"}
```

Exporting an LTPA key file with web service

To export an LTPA key file with the RESTful web service, issue an HTTP GET command and include the `export` request parameter on the LTPA Keys resource URI.

URL

```
https://{appliance_hostname}/ltpa_key/{id}?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the LTPA key file to export.

Response

HTTP response code and the LTPA key file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/ltpa_key/{id}?export
```

Response:

```
200 ok

The LTPA key file
```

Renaming an LTPA key file with web service

To rename an LTPA key file with the RESTful web service, issue an HTTP POST command on the LTPA Keys resource URI.

URL

```
https://{appliance_hostname}/ltpa_key/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	The existing LTPA Key File name.
new_name	The new LTPA Key File name.

Response

HTTP response code and JSON data that represents the new LTPA key file name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/ltpa_key/{id}

PUT_DATA: {
"new_name": "new_ltpa_key_name"
}
```

Response:

```
200 ok

{"id":"The new LTPA key file name"}
```

Deleting an LTPA key file with web service

To delete an LTPA key file with the RESTful web service, issue an HTTP DELETE command on the LTPA Keys resource URI.

URL

```
https://{appliance_hostname}/ltpa_key/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the LTPA key file to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/ltpa_key/{id}
```

Response:

```
200 ok
```

Query site contents

You can manage query site contents files with either the local management interface or the web service.

Managing query site contents files with local management interface

To manage query site contents files with the local management interface, use the Query Site Contents management page.

Procedure

1. From the top menu, select **Secure Reverse Proxy Settings** > **Tools** > **Query Site Contents**. All query contents files and their version information are displayed.
2. Select the file of interest.
3. Click **Export** if you want to export the file to your local drive.

   Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
4. Confirm the save operation in the browser window that pops up.

Retrieving all query site contents file names with web service

To retrieve all query site contents file names with the RESTful web service, issue an HTTP GET command on the query site contents resource URI.

URL

```
https://{appliance_hostname}/query_sitecontents
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

GET

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON date representing the name, version, and description of the query site contents files

Parameter	Description
id	The query site contents file name.
version	The current version of file. This information is the last modified time of the file. A numerical number that indicates the number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).
description	A description of the file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/query_sitecontents
```

Response:

```
200 ok

[
{
"id":"file_1",
"version":"version_id"
"description":"A description of the file"
},
{
"id":"file_2",
"version":"version_id"
"description":"A description of the file"
},
...
]
```

Retrieving the contents of a query site contents file with web service

To retrieve the contents of a query site contents file with the RESTful web service, issue an HTTP GET command on the query contents resource URI.

URL

```
https://{appliance_hostname}/query_sitecontents/{id}
```

Note:

- The web service request must use the format that is described in "Required header for calling a web service" on page 10.
- This web service cannot be used for a binary query site contents file.

Method

`GET`

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the query site contents file to retrieve.

Response

HTTP response code and JSON data that represents content of the query site contents file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/query_sitecontents/{id}
```

Response:

```
200 ok

{"contents":"The contents of the query site contents file"}
```

Exporting a query site contents file with web service

To export the contents of a query site contents file with the RESTful web service, issue an HTTP GET command and include the `export` parameter on the query contents resource URI.

URL

```
https://{appliance_hostname}/query_sitecontents/{id}?export
```

The `export` parameter is what differentiates the retrieve operation and the export operation. If the `export` parameter is included in the URL, the entire file is exported. If no `export` parameter is included in the URL, JSON data that represents the file contents is retrieved.

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

`GET`

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
id	Name of the query site contents file to export.

Response

HTTP response code and the query site contents file.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/query_sitecontents/{id}?export
```

Response:

```
200 ok

The query site contents file
```

Chapter 7. Manage: System Settings

Information about configuring Security, Network, and System settings of your appliance.

Updates and licensing

Information about managing updates and licensing on your Security Web Gateway Appliance.

Viewing the update and licensing overview

The Overview page displays current information about the Security Web Gateway Appliance firmware, intrusion prevention content, update servers, and licenses.

About this task

You can install available updates on the Overview page. However, to manually add updates, check for updates, and schedule updates, you must visit the Available Updates page.

Procedure

1. Click **Manage System Settings** > **Updates and Licensing** > **Overview**.
2. Perform one or more of the following actions:
   - To install an available update, click **Install**.
   - To register a license, click **Register License**.

     Note: The Register License link displays only when you do not have a registered license.
     a. On the Licensing page, click **Select License** and locate the license file that you want to install.
     b. Select the license file that you want to install and then click **Open**.
     c. Click **Save Configuration**.

Installing updates

Install firmware and intrusion prevention updates to improve the Security Web Gateway Appliance and the network protection that is provided by the appliance.

About this task

Important: After you install firmware updates, you must restart the appliance.

Firmware updates contain new program files, fixes or patches, enhancements, and online help.

Intrusion prevention updates contain the most recent security content that is provided by IBM X-Force research and development team.

Procedure

1. Click **Manage System Settings** > **Updates and Licensing** > **Available Updates**.
2. On the Available Updates page, use one or more of the following commands:

Option	Description
Upload	To manually add an update, click **Upload**. In the New Update window, click **Select Update**, browse to the update file, click **Open**, and then click **Submit**. **Note:** You can install the update after you manually add it.
Refresh	To check for updates, click **Refresh**.
Install	To install an update, select the update, and then click **Install**.
Schedule	To create or edit an update schedule, select an update, and then click **Schedule**. In the Edit Schedule window, take one or more of the following actions: • To remove an update schedule, select **Remove Schedule**. • To create an update schedule, select a date and time to install the update. Click **Submit** to save your changes.

Configuring the update schedule

Configure the update schedule to receive X-Force content updates daily, weekly, or according to a specified interval of time.

About this task

A 15-minute buffer is applied to update times so that update servers do not become overburdened. Updates are downloaded up to 15 minutes before or after the time you specify.

Procedure

1. Click **Manage System Settings** > **Updates and Licensing** > **Update Schedule**.
2. In the Update Schedule pane, select **Auto Update** to receive X-Force content updates.
3. Use one of the following methods to schedule updates:
   - To receive updates on a daily basis, select **Daily or Weekly**, select **Every Day** from the first list, and then select a time from the second list.
   - To receive updates on a weekly basis, select **Daily or Weekly**, select the day of the week you would like to receive updates on, and then select a time from the second list.
   - To receive updates on a schedule that ranges from 1 hour to 24 hours, select **Specified Interval**, and then select the update interval in minutes.

     Range: 60 - 1440 minutes
4. Click **Save**.

Configuring update server settings

Configure your appliance to download update files from an update server.

About this task

You can configure multiple, ordered servers for failover.

Note: You cannot delete the IBM ISS Default License and Update Server. You can disable it.

Procedure

1. Click **Manage System Settings** > **Updates and Licensing** > **Update Servers**.
2. In the Update Servers pane, take one of the following actions:
   - To add an update server, click **New**. The Add Server window is displayed.
   - To edit an update server, select the server, and then click **Edit**. The Edit Server window is displayed.
   - To delete an update server, select the server, and then click **Delete**.
3. When you add or edit an update server, configure the following options on the General tab:

Option	Description
Order	Defines the order in which update servers are queried for appliance software updates. The appliance uses the next server on the list when a server takes more than 24 hours to respond.
Enable	Enables the update server so that it can be used by the appliance.
Name	A name that describes the update server.
Server Address	The IP address or DNS name of the update server.
Port	The port number that the appliance uses to communicate with the update server. **Tip:** The port number for the IBM ISS Download Center is 443. The default port for internal update servers is 3994.

Option	Description
Trust Level	Defines how the appliance is authenticated with the update server. **Explicit (user-defined)** The appliance uses the local certificate that is pasted into the Certificate box to authenticate the connection to the update server. The certificate must be Base64 PEM-encoded data. Explicit trust is the most secure trust level. Explicit trust certificates must be Base64 PEM-encoded data. **Explicit (xpu.iss.net)** The appliance uses the local certificate for the IBM ISS update server to authenticate the connection to the update server. The IBM ISS update server certificate is installed on the appliance by default. The certificate is Base64 PEM-encoded data. Explicit trust is the most secure trust level. Explicit trust certificates must be Base64 PEM-encoded data. **First Time Trust** If a certificate is not on the appliance, the appliance downloads a certificate from the server when it connects to the server for the first time. First Time Trust is more secure than Trust All and less secure than Explicit Trust. **Note:** After the appliance downloads the certificate, it reverts to explicit-trust functionality. **Trust All** The appliance trusts the update server, and does not use SSL certificates for authentication. Trust all trust is the least secure trust level. **Attention:** The Trust All trust level presents a security risk because the internal update server can be spoofed and redirected to a fake server.

4. Optional: If you use a proxy server, configure the following settings on the Proxy Settings tab:

Option	Description
Use Proxy	Enables the appliance to use a proxy server for update servers.
Server Address	The IP address or DNS name of the proxy server. **Note:** The **Server Address** field is displayed when you select the Use Proxy check box.
Port	The port number that the proxy server uses to communicate with the update server. **Note:** The **Port** field is displayed when you select the Use Proxy check box.
Use Authentication	Enables the appliance to authenticate to a proxy server.
User Name	User name that is required for authenticating to the proxy server. **Note:** The **User Name** field is displayed when you select the Use Authentication check box.
Password	Password that is required for authenticating to the proxy server. **Note:** The **Password** field is displayed when you select the Use Authentication check box.

5. Click **Submit**.

Viewing update history

View the update history to see which firmware and security content updates are downloaded, installed, and rolled back on the appliance.

About this task

After you install an update, the update package is deleted from the appliance.

Procedure

1. Click **Manage System Settings** > **Updates and Licensing** > **Update History**.
2. To refresh the page, click **Refresh**.

Installing a fix pack

Install a fix pack when IBM Customer Support instructs you to do so.

Before you begin

The appliance does not automatically create a backup copy of a partition when you apply a fix pack to it. If you want to back up your partition before you apply the fix pack, then you must do it manually.

Restriction: You cannot uninstall fix packs.

About this task

Fix packs are applied to the current partition. If a fix pack is installed on your appliance, you can view information about who installed it, comments, patch size, and the installation date.

Procedure

1. Click **Manage**, and then click **Fix Packs**.
2. In the Fix Packs pane, click **New**.
3. In the Add Fix Pack window, click **Browse** to locate the fix pack file, and then click **Open**.
4. Click **Submit** to install the fix pack.

Installing a license

You must install a current license file to receive updates to the appliance.

About this task

Contact your IBM representative to get a license registration number. You can download and register your license from the IBM Security Systems License Key Center at https://ibmss.flexnetoperations.com.

Procedure

1. Optional: If you are not configuring your appliance for the first time, click **Manage** > **Licensing**.
2. On the Licensing page, click **Select License** and locate the license file that you want to install.
3. Select the license file that you want to install and then click **Open**.
4. Click **Save Configuration**.

   Note: OCNID stands for Order Confirmation Number and ID.

Managing firmware settings

The IBM Security Web Gateway Appliance has two partitions with separate firmware on each partition. Partitions are swapped during firmware updates, so that you can roll back firmware updates.

About this task

Either partition can be active on the appliance. In the factory-installed state, partition 1 is active and contains the firmware version of the current released product. When you apply a firmware update, the update is installed on partition 2 and your policies and settings are copied from partition 1 to partition 2. The appliance restarts the system using partition 2, which is now the active partition.

Note: The appliance comes with identical firmware versions installed on both of the partitions so that you have a backup of the initial firmware configuration.

Tip: Avoid swapping partitions to restore configuration and policy settings. Use snapshots to back up and restore configuration and policy settings.

Procedure

1. Click **Manage System Settings** > **Updates and Licensing** > **Firmware Settings**.
2. On the Firmware Settings page, perform one or more of the following actions:

Option	Description
Edit	To edit the comment that is associated with a partition, select the partition and click **Edit**.
Create Backup	**Important:** Create a backup of your firmware only when you are installing a fix pack that is provided by IBM Customer Support.Fix packs are installed on the active partition and you might not be able to uninstall the fix pack. **Note:** The backup process can take several minutes to complete.
Set Active	Set a partition active when you want to use the firmware that is installed on that partition. For example, you might want to set a partition active to use firmware that does not contain a recently applied update or fix pack.

3. Click **Yes**. If you set a partition active, the appliance restarts the system using the newly activated partition.

Network Settings

Information about configuring network interfaces and information about your Security Web Gateway Appliance.

Application interface management

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage application interfaces.

Managing application interfaces with local management interface

To manage application interfaces with the local management interface, use the Application Interfaces management page.

Procedure

1. From the top menu, select **Manage System Settings** > **Network Settings** > **Application Interfaces**. All current application interfaces are displayed in tabs. Each tab contains the current addresses and settings for a particular interface.
2. Select the tab of the interface that you want to work with. You can then add, edit, or delete an address on the corresponding interface tab.
   - **Add an address**
     a. Click **New**.
     b. In the Add Address page, provide details of the address to add.
        – Select the **Enabled** check box if you want this address to be enabled after creation.
        – Select **IPv4** or **IPv6** to indicate the type of address to add.
          - If **IPv4** is selected:
            1) Under **IPv4 Settings**, select either **Static** or **Auto** to indicate whether the IPv4 address is static or DHCP-assigned.

Note: Only one address per interface can be set to auto. If an existing address is already set to auto, then the **Auto** check box is disabled.

2) *Optional*: If **Static** is selected in the previous step, you must enter the IPv4 address and subnet mask. If **Auto** is selected in the previous step, you can ignore the **Address** and **Subnet Mask** field.

- If **IPv6** is selected, enter the IPv6 **Address** and **Prefix**.

– Click **Save**.

- **Modify an address**
  – *Method 1:*
    a. Select the address to modify from the table.
    b. Click **Edit**.
    c. In the Edit Address page, modify as needed. See the "Add an address " section for descriptions of the fields.
    d. Click **Save** to save your changes.
  – *Method 2:*
    a. In the table, double-click the field to edit.
    b. Make changes inline.

    Note: Only some fields can be edited inline.
    c. Click outside the editing field to save the changes.
- **Delete an address**
  a. Select the address to delete from the table.
  b. Click **Delete**.
  c. In the Delete Address page, click **Yes** to confirm the deletion.
- **Test connection to a server**
  a. Click **Test**.
  b. On the Ping Server page, enter the IP address or name of the server to test the connection with.
  c. Click **Test**. A message is then displayed indicating whether the ping operation was successful.

Retrieving the list of application interfaces with web service

To retrieve the list of application interfaces with the web service, issue an HTTP GET command on the application interfaces resource URI.

URL

```
https://{appliance_hostname}/application_interfaces
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the name and state of the application interfaces.

Parameter	Description
id	Name of the application interface. Valid values are “P.1”, “P.2”, “P.3”, and “P.4”.
state	State of the application interface. Valid values are “up”, “down”, and “disabled”.

Note: For descriptions of possible error responses that can be returned from a web service call, see “Web service responses” on page 11.

Example

Request:

```
GET https://{appliance_hostname}/application_interfaces
```

Response:

```
200 ok

[
{
"id":"P.1",
"state":"up/down/disabled"
},
{
"id":"P.2",
"state":"up/down/disabled"
},
{
"id":"P.3",
"state":"up/down/disabled"
},
{
"id":"P.4",
"state":"up/down/disabled"
},
]
```

Retrieving the list of address identifiers for an interface with web service

To retrieve the list of address identifiers for an interface with the RESTful web service, issue an HTTP GET command on the application interfaces resource URI.

URL

```
https://{appliance_hostname}/application_interfaces/{interface_id}/addresses
```

Note: The web service request must use the format that is described in “Required header for calling a web service” on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
interface_id	ID of the interface to retrieve the list of addresses for. Valid values are `P.1`, `P.2`, `P.3`, and `P.4`.

Response

HTTP response code and JSON data that represents the list of addresses (unique identifiers).

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/application_interfaces/{interface_id}/addresses
```

Response:

```
200 ok

[
"1","2","3","4"
]
```

Each address that is created on an interface is assigned a unique identifier. The identifier can then be used as a reference for this address in future requests. The `"1","2","3","4"` value that is mentioned earlier represents such unique identifiers.

Retrieving the current settings for an address with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the application interface addresses resource URI.

URL

```
https://{appliance_hostname}/application_interfaces/{interface_id}/addresses/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
interface_id	ID of the interface to which the address is attached. Valid values are `P.1`, `P.2`, `P.3`, and `P.4`.
id	ID of the address to retrieve settings for. Valid values are `0-98`.

Response

HTTP response code and JSON data that represents the addresses and settings.

Parameter	Description
type	Indicates whether the address is an IPv4 address or an IPv6 address. The value is `ipv4` or `ipv6`.
enabled	Indicates whether the address is enabled. The value is `true` or `false`.
mode4	Indicates whether the address is static or DHCP-assigned. The value is `static` or `auto`. This field is only returned if the address type is `ipv4`.
netmask	If the **mode4** parameter is set to `static`, this field contains the user specified IPv4 subnet mask. If the **mode4** parameter is set to `auto`, this field contains the DHCP-assigned IPv4 subnet mask. **Note:** Because there is a lag between configuring an address and DHCP assigning the address value, this field might be empty for auto addresses. This field is only returned if the address type is `ipv4`.
prefix	This field contains the user-specified IPv6 prefix. This field is only returned if the address type is `ipv6`.
address	If the **type** parameter is set to `ipv4` and the **mode4** parameter is set to `static`, this field contains the user-specified IPv4 address. If the **type** parameter is set to `ipv4` and the **mode4** parameter is set to `auto`, this field contains the DHCP-assigned IPv4 address. **Note:** Because there is a lag between configuring an address and DHCP assigning the address value, this field might be empty for auto addresses. If the address type is IPv6, this field contains the user-specified IPv6 address.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/application_interfaces/{interface_id}
/addresses/{id}
```

Response:

```
200 ok

{
"id":"The address identifier",
"type":"The address type (ipv4/ipv6)",
"enabled":"true/false",
"mode4":"static/auto",
```

```
"netmask":"The IPv4 subnet mask",
"prefix":"The IPv6 prefix",
"address":"The IP address"
}
```

Listing IP addresses for all application interfaces with web service

To list IP addresses for all application interfaces with the web service, issue an HTTP GET command on the Reverse Proxy application interface resource URI.

URL

```
https://{appliance_hostname}/wga_templates/ipaddress
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the list of IP addresses for all application interfaces.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/ipaddress
```

Response:

```
200 ok

[
{"id":"IPv6 Address"},
{"id":"IPv6 Address"},
{"id":"IPv4 Address"},
{"id":"IPv4 Address"}
]
```

Testing connection to a server with web service

To ping a server with the RESTful web service, issue an HTTP POST command on the application interfaces resource URI.

URL

```
https://{appliance_hostname}/application_interfaces/ping
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

`POST`

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
server	Specifies the IP address or name of the server to ping. This parameter is required.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/application_interfaces/ping

POST_DATA:
{
"server":"server to ping"
}
```

Response:

```
200 ok
```

Adding an address to an interface with web service

To add an address to an interface with the RESTful web service, issue an HTTP POST command on the application interface addresses resource URI.

URL

`https://{appliance_hostname}/application_interfaces/{interface_id}/addresses`

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

`POST`

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
interface_id	ID of the interface to add address to. Valid values are `P.1`, `P.2`, `P.3`, and `P.4`.
enabled	Indicates whether the address is enabled. This parameter is required. The value is `true` or `false`.

Parameter	Description
mode4	Indicates whether the IPv4 address is static or DHCP-assigned. This parameter is required if type is set to **ipv4**. The value is `static` or `auto`. **Note:** Only 1 address per interface can be auto. All others must be static.
address	If the **type** parameter is set to `ipv4` and the **mode4** parameter is set to `auto`, this field is ignored. Otherwise this field is required. If the **type** parameter is set to `ipv4` and the **mode4** parameter is set to `static` or the **type** is set to `ipv6`, this field contains the user specified IP address.
netmask	If the **type** parameter is set to `ipv4` and the **mode4** parameter is set to `static`, this field is required and must contain the IPv4 subnet mask.
type	This parameter is required. The value is `ipv4` or `ipv6`.
prefix6	This parameter contains the IPv6 prefix.

Response

HTTP response code and JSON data that represents details of the address added.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

To create an IPv4 address:

```
POST https://{appliance_hostname}/application_interfaces/{interface_id}/addresses

POST_DATA:
{
"enabled":"true/false",
"mode4":"static/auto",
"address":"ipv4 address",
"netmask":"subnet mask",
"type":"ipv4"
}
```

To create an IPv6 address:

```
POST https://{appliance_hostname}/application_interfaces/{interface_id}/addresses

POST_DATA:
{
"enabled":"true/false",
"address":"ipv6 address",
"prefix":"ipv6 prefix",
"type":"ipv6"
}
```

Response:

```
200 ok

{
"id":"unique_id",
"type":"ipv4/ipv6",
"enabled":"true/false",
"mode4":"static/auto",
"address":"ip address",
"netmask":"subnet mask",
"prefix":"ipv6 prefix"
}
```

Modifying a current address on an interface with web service

To modify a current address on an interface with the web service, issue an HTTP PUT command on the application interface addresses resource URI.

URL

```
https://{appliance_hostname}/application_interfaces/{interface_id}/addresses/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
interface_id	ID of the interface to modify address settings for. Valid values are `P.1`, `P.2`, `P.3`, and `P.4`.
enabled	Indicates whether the address is enabled. This parameter is required. Valid value is "true" or "false".
mode4	Indicates whether the IPv4 address is static or DHCP-assigned. This parameter is required if type is set to **ipv4**. Valid value is "static" or "auto". **Note:** Only 1 address per interface can be auto. All others must be static.
address	If the address is IPv6 or the **mode4** parameter is set to `static`, this field contains the IP address. If the **mode4** parameter is set to `auto`, this field is ignored. This parameter is optional. If no value is specified, the current value remains. If the current value of the **mode4** parameter is changed from `auto` to `static`, this parameter is required.

Parameter	Description
netmask	This field contains the IPv4 subnet mask. If the **mode4** parameter is set to auto, this field is ignored. This parameter is optional. If no value is specified, the current value remains. If the current value of the **mode4** parameter is changed from auto to static, this parameter is required.
type	This parameter is required.
prefix	This parameter is the IPv6 prefix. This parameter is optional.

Response

HTTP response code and JSON data that represents the address and include a unique identifier.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

Modify an IPv4 address

```
PUT https://{appliance_hostname}/application_interfaces/{interface_id}
/addresses/{id}

PUT_DATA:
{
"enabled":"true/false",
"mode4":"static/auto",
"address":"ipv4 address",
"netmask":"subnet mask",
}
```

Modify an IPv6 address

```
PUT https://{appliance_hostname}/application_interfaces/{interface_id}
/addresses/{id}

PUT_DATA:
{
"enabled":"true/false",
"address":"ipv6 address",
"prefix":"ipv6 prefix",
}
```

Response:

```
200 ok

{
"id":"unique_id",
"type":"ipv4/ipv6",
"enabled":"true/false",
"mode4":"static/auto",
```

```
"address":"ip address",
"netmask":"subnet mask",
"prefix":"ipv6 prefix"
}
```

Deleting a current address from an interface with web service

To delete an address from an interface with the RESTful web service, issue an HTTP DELETE command on the application interface addresses resource URI.

URL

```
https://{appliance_hostname}/application_interfaces/{interface_id}/addresses/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
interface_id	ID of the interface to delete address settings for. Valid values are `P.1`, `P.2`, `P.3`, and `P.4`.
id	A unique identifier that is generated for each address on an application interface. Valid values are `0-98`.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/application_interfaces/{interface_id}
/addresses/{id}
```

Response:

```
200 ok
```

Finding the next available HTTPS port for a particular application interface with web service

To find the next available HTTPS port for a particular application interface, issue an HTTP GET command on the Reverse Proxy application interface resource URI.

URL

```
https://{appliance_hostname}/wga_templates/httpsport/{ip_addr}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

`GET`

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
ip_addr	The IPv6 or IPv4 address of the application interface. For example, `2002:930:a006:164:20c:29ff:fe41:1c49` or `9.48.164.244`.

Response

HTTP response code and JSON data that represents the HTTPS port.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/httpsport/{ip_addr}
```

Response:

```
200 ok

[{"id":"HTTPS PORT#"}]
```

Finding the next available HTTP port for a particular application interface with the web service

To complete this operation with the web service, issue an HTTP GET command on the application interface resource URI.

URL

`https://{appliance_hostname}/wga_templates/httpport/{ip_addr}`

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

`GET`

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
ip_addr	The IPv6 or IPv4 address of the application interface. For example, `2002:930:a006:164:20c:29ff:fe41:1c49` or `9.48.164.244`.

Response

HTTP response code and JSON data that represents the HTTP port.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/wga_templates/httpport/{ip_addr}
```

Response:

```
200 ok

[{"id":"HTTP PORT#"}]
```

Configuring management interfaces

Use the Management Interfaces page to view and manage the network security interfaces for the appliance.

About this task

Note: If you change the IP address of the management interfaces, connect your web browser to the new IP address for future sessions.

Procedure

1. Click **Manage System Settings** > **Network Settings** > **Management Interfaces**.
2. On the Management Interfaces page, type a **Host name**.
3. To enable network users to locate the appliance using zero configuration networking, select **Advertise managment interface using multicast DNS**.
4. Select the **Default Interface**.
5. To enable the other management interface, select **Enable** *interface name*.
6. Click the tab for the primary interface, and then click **IPV4** or **IPV6**.
7. Configure the following options:

Option	Description
Auto/Manual	Select **Auto** to acquire an IP address from a DHCP server. Select **Manual** to specify a static IP address, Netmask, and Gateway (IPv4) or Prefix (IPv6).
Address	If you selected **Manual** mode, type the IP address that you want to use for the interface.
Gateway	If you selected **Manual** mode, type the Gateway for the interface.
Netmask (IPv4)	If you selected **Manual** mode for IPv4, type the Subnet Mask for the interface.
Prefix (IPv6)	If you selected **Manual** mode for IPv6, type the prefix length for the interface.

8. Click the DNS tab, and then configure the following options:

Option	Description
Auto/Manual	Select **Auto** to acquire DNS server addresses from a DHCP server. Select **Manual** to specify DNS servers.
Primary DNS	Specifies the primary DNS server IP address.
Secondary DNS	Specifies the secondary DNS server IP address.
Tertiary DNS	Specifies an optional third DNS server IP address.
DNS Search Path	Specifies one or more DNS search paths. Separate each path with a comma.

9. Click the tab for the secondary interface, and then click **IPV4** or **IPV6**.
10. Configure the following options:
    - Auto/Manual
    - Address
    - Gateway
    - Netmask (IPv4)
    - Prefix (IPv6)
11. Click **Save**.

Configuring static routes

Configure static routes to the paired protection interfaces on your appliance to enable network routers to redirect users to block pages or authentication pages.

About this task

This task is only necessary for networks that contain an additional network segment between the user segment and the appliance.

Procedure

1. Click **Manage System Settings** > **Network Settings** > **Static Routes**.
2. On the Static Routes page, take one of the following actions:
    - Click **New** to create a route.
    - Select an existing route, and then click **Edit**.
3. Define the following information in each field:
    - Destination
    - Gateway
    - Metric
    - Interface or Segment
4. Click **Save**.

Front-end load balancer

The appliance provides front-end load balancing function to automatically assign client requests to the appropriate reverse proxy server based on the scheduling specified algorithm.

In a typical setup, there are two front-end load balancer servers and multiple reverse proxy servers. A front-end load balancer is a server that uses a virtual IP

address to accept requests from a client, determines which reverse proxy server is most suitable based on the specified scheduling algorithm, and forwards the requests to the appropriate reverse proxy server. A heartbeat is transmitted between the two front-end load balancers so that the state of each front-end load balancer is known. If the primary front-end load balancer is unavailable and thus no heartbeat can be detected, the backup load balancer assumes the virtual IP address of the primary load balancer and start accepting requests from the client.

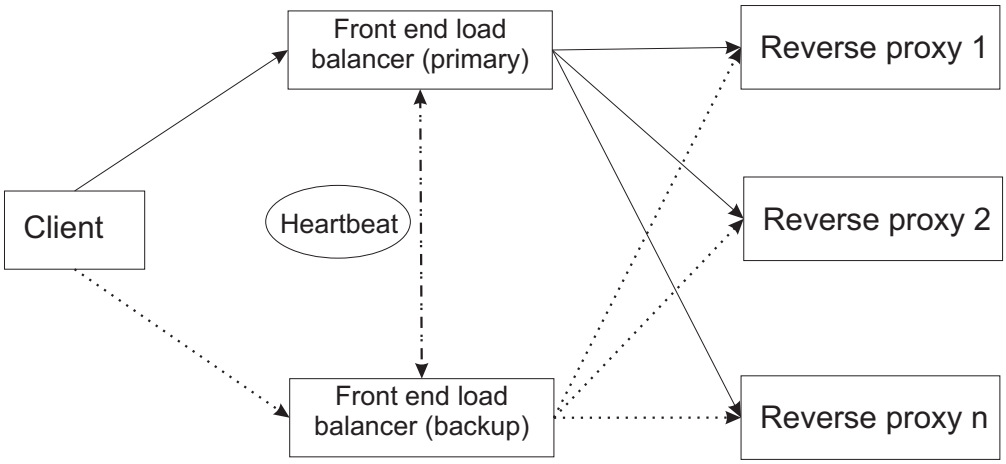

Figure 1. Front-end load balancer

Note: You can have only two front-end load balancers in your environment.

It is possible to configure the reverse proxy functionality on a machine that is also acting as a front-end load balancer. However, this might have a negative impact on the performance of the front-end load balancer. If you decide to use such setting, you must take the resource that is consumed by the reverse proxy into consideration. Make sure that the front-end load balancer still has enough resources to perform routing effectively.

Note: The back-end IP address must be set as the default gateway for load balanced servers. After enabling the front-end load balancer, ensure that each of the back-end servers set their default gateway as the value set in the back-end **Address** field.

Scheduling

The front-end load balancing function of the appliance supports several types of scheduling.

The supported scheduling types are:

lc Least connection

rr Round robin

wlc Weighted least connection

wrr Weighted round robin

lblc Locality-based least connection

dh Destination hashing

sh Source hashing

Persistence

Persistence is a mechanism that ensures a client is connected to the same reverse proxy server during a session.

The persistence functionality acts at the TCP layer (Layer 4). It can be enabled or disabled through configuration.

Persistence is controlled by the client IP address and the destination port number.

Configuring front-end load balancer with local management interface

To configure the front end load balancer with the local management interface, use the Front End Load Balancer management page.

Procedure

1. From the top menu, select **Manage System Settings** > **Network Settings** > **Front End Load Balancer**.
2. *Optional:* Expand **View Diagram** to see a network diagram that helps associate the various terms that are used within the configuration panels to the corresponding location within the network.
3. On the **General** tab page:
   a. Select **Enabled** if you want to enable this front-end load balancer.
   b. Select **Debug** if you want more debug messages to be sent to the security log.
   c. Under **Gateway**:

      Note: This is the network that the load balancer and the load balanced servers communicate over.

      1) For the **Gateway Address** field, specify the IP address that connects this front-end load balancer to the private network.

         Note: The back-end address must be set as the default gateway for load balanced servers. After enabling the front-end load balancer, ensure that each of the back-end servers set their default gateway as the value set in the back-end **Address** field that is mentioned previously.

      2) For the **Mask** field, specify the network mask for the subnet that connects this front-end load balancer to the private network.
      3) Select the back-end interface from the list under **Interface**.
   d. *Optional:* Click **Event log** to view system events.
4. On the **Servers** tab page, you can work with virtual servers and real servers. Each virtual server corresponds to an interface (virtual IP address and port) that is load balanced. Each real server corresponds to a load balanced server.
   - **Add a virtual server**
     a. Click **New**.
     b. On the Add Virtual Server page, define settings of the virtual server to be added.

        On the **General** tab page:

Field	Description
Enabled	Specifies whether the new virtual server is active.
Name	Name of the virtual server.

Field	Description
Virtual Address	Specifies the IP address that connects this virtual server to the public network.
Port	Specifies the port on which this virtual server listens.
Mask	Specifies the network mask to be applied to the IP address for the virtual server.
Interface	Specifies the appliance interface on which the new virtual server connects to the public network.

On the **Scheduler** tab page:

Field	Description
Scheduler	Specifies the scheduling algorithm for distributing jobs to the real servers. Available choices are: • Round-Robin • Weighted Round-Robin • Least-Connection • Weighted Least-Connection • Locality-Based Least-Connection • Destination Hash • Source Hash
Timeout	Specifies the time in seconds that a virtual server must be inactive before it is removed from the routing table.
Re-Entry	Specifies the time in seconds that a virtual server must be inactive before it is added back to the routing table after being previously removed because of failure.
Persistence	Specifies in seconds the time to allow a persistent virtual server connection to remain active. If the value is missing or set to 0, persistence is turned off.

c. Click **Save**.

- **Delete a virtual server**
  a. Select the virtual server to delete from the list.
  b. Click **Delete**.
  c. On the confirmation page, click **Yes**.
- **Edit a virtual server**
  – Method 1:
    a. Select the virtual server to edit from the list.
    b. Click **Edit**.
    c. On the Edit Virtual Server page, modify the settings as needed.
    d. Click **Save**.
  – Method 2:
    a. Double-click the field to edit.

       Note: All fields except **Name** can be edited inline.
    b. Make changes inline.
    c. Click outside the editing field to save the changes.
- **Manage real servers**

a. From the list of virtual servers, select the virtual server to associate the real servers with.
b. Click **Real Servers**. The Real Servers page is displayed.
   – To add a real server:
     1) Click **New**.
     2) On the Add Real Server page that pops up, define settings for the reverse proxy server to be added.

Field	Description
Enabled	Specifies whether the new real server is active.
Address	Specifies the IP address for the real server.
Weight	Specifies an integer that represents this processing capacity of the server relative to that of other real servers. For example, a server assigned 2000 has twice the capacity of a sever assigned 1000. The weighted scheduling algorithms adjust this number dynamically based on workload.

     3) Click **Save**.
   – To delete a real server:
     1) Select the real server to delete from the list.
     2) Click **Delete**.
     3) On the confirmation page, click **Yes**.
   – To edit a real server:
     - Method 1:
       1) Select the real server to edit from the list.
       2) Click **Edit**.
       3) On the Edit Real Server page, modify the settings as needed.
       4) Click **Save**.
     - Method 2:
       1) Double-click the field to edit.

          Note: All fields except **Address** can be edited inline.
       2) Make changes inline.
       3) Click outside the editing field to save the changes.
c. Click **Close** to return to the Front End Load Balancer main page.

5. On the **High Availability** tab page, you can define the settings that enable high availability of the front-end load balancer function. For example, configure a second front-end load balancer as either a primary or a back-up load balancer for the environment.
   a. Select the **Enable High Availability** check box to enable this feature.
   b. Select **Primary** or **Backup** to designate this system as the primary or backup front-end load balancer.
   c. For the **Local Address - Primary** field, select the local IP address of the front-end load balancer.
   d. For the **Remote Address - Backup** field, specify the IP address that is used by this system to communicate with the other front-end load balancer. This field is required if a backup load balancer is in use.

e. In the **Health Check Interval** field, specify in seconds the interval of the heartbeat messages that are sent between the primary and backup front-end load balancers.

f. In the **Health Check Timeout** field, specify in seconds the time to wait before the system declares a non-responsive router unavailable and initiating failover.

6. Click **Save** to save all changes that are made on the Front End Load Balancer management page.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Retrieving front-end load balancer configuration in full with web service

To retrieve the current front-end load balancer configuration with the RESTful web service, issue an HTTP GET command on the FELB file resource URI.

URL

```
https://{appliance_hostname}/felb
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the front-end load balancer configuration.

Table 6. Front-end load balancer level attributes

Parameter	Description
enabled	Specifies whether the front-end load balancer is enabled. The value is `true` or `false`.
debug	Designates whether additional debug messages are sent to the security log file. The value is `true` or `false`.
is_primary	Specifies whether this system is the primary front-end load balancer or the backup. The value is `true` or `false`.
local	Specifies the local IP address of the front-end load balancer.
remote_active	Specifies whether the remote front-end load balancer is active. The value is `true` or `false`.

Table 6. Front-end load balancer level attributes (continued)

Parameter	Description
remote	Specifies the IP address which is used by this system to communicate with the other front-end load balancing system. This parameter is required if a back-up load balancer is in use.
keepalive	Specifies the interval, in seconds, between the heartbeat messages that are sent between the front-end load balancing systems.
deadtime	Specifies the number of seconds to wait before the system declares a non-responding router dead and initiating failover.
nat_router	Specifies the appliance interface on which the front-end load balancer connects to the private network.
nat_nmask	Specified the network mask for the subnet which connects this system to the private network.
nat_interface	Specifies the appliance interface on which the front-end load balancer connects to the private network. The value must one of the management interface names (M.1, M.2) or application interface names (P.1, P.2, P.3, P.4).

Table 7. Service level attributes

Parameter	Description
active	Specifies whether the new service is active. The value is `true` or `false`. This parameter is required.
name	Specifies the name of the new service. This parameter is required.
address	Specifies the IP address that connects this service to the public network. This parameter is required.
vip_nmask	Specifies the network mask to be applied to the IP address for the service. This parameter is required.
timeout	Specifies the time in seconds that a service must be inactive before it is removed from the routing table. This parameter is required.
reentry	Specifies the time in seconds that a service must be active before it is added back into the routing table after it is previously removed because of failure. This parameter is required.
port	Specifies the port on which this service listens. This parameter is required.

Table 7. Service level attributes (continued)

Parameter	Description
scheduler	Specifies the scheduling algorithm for distributing jobs to the real servers. The choices are: **lc** Least connections **rr** Round robin **wlc** Weighted least connections **wrr** Weighted round robin **lblc** Locality based least connection This parameter is required.
persistent	Specifies the time (in seconds) to allow a persistent service connection to remain active. If this value is missing or set to 0, persistence is turned off. This parameter is required.
interface	Specifies the appliance interface on which the new service connects to the public network. The value must one of the application interface names (P.1, P.2, P.3, P.4). This parameter is required.
serverCount	The number of real servers that are attached to this service.

Table 8. Server level attributes

Parameter	Description
active	Specifies whether the new server is active. This parameter is required.
address	Specifies the IP address for the real server. This parameter is required.
weight	Specifies an integer that represents the processing capacity of this server relative to that of other real servers. For example, a server assigned 2000 has twice the capacity of a server assigned 1000. The weighted scheduling algorithms adjust this number dynamically based on workload. This parameter is required.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/felb
```

Response:

```
200 ok

{
"enabled":"true/false",
"is_primary":"true/false",
```

```
"debug":"true/false",
"remote_active":"true/false",
"local":"ip address",
"remote":"ip address",
"keepalive":"integer",
"deadtime":"integer",
"nat_router":"ip address",
"nat_interface":"interface id",
"nat_nmask":"subnet mask",
"services":[{
            "active":"true/false",
            "name":"service name",
            "address":"ip address",
            "vip_nmask":"subnet mask",
            "timeout":"integer",
            "reentry":"integer",
            "port":"port number",
            "scheduler":"lc/rr/wlc/wrr/lblc/dh",
            "persistent":"integer",
            "interface":"interface id",
            "serverCount":"integer",
            "servers":[{
                        "active":"true/false",
                        "address":"ip address",
                        "weight":"integer"
                       }]
          }]
}
```

Retrieving front-end load balancer level configuration attributes with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the FELB file resource URI.

URL

```
https://{appliance_hostname}/felb/configuration
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the front-end load balancer level configuration attributes.

These parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/felb/configuration
```

Response:

```
200 ok

{
"enabled":"true/false",
"is_primary":"true/false",
"debug":"true/false",
"remote_active":"true/false",
"local":"ip address",
"remote":"ip address",
"keepalive":"integer",
"deadtime":"integer",
"nat_router":"ip address",
"nat_interface":"interface id",
"nat_nmask":"subnet mask"
}
```

Retrieving services of the front-end load balancer with web service

To retrieve services of the front-end load balancer with the web service, issue an HTTP GET command on the FELB service resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the configured service names.

Parameter	Description
id	The name of the configured service for the front-end load balancer

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/felb/configuration/services
```

Response:

```
200 ok

[
{"id":"service name1"},
{"id":"service name2"}
...
]
```

Retrieving service configuration attributes with web service

To complete the operation with the RESTful web service, issue an HTTP GET command on the FELB service resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the service to retrieve configuration attributes for

Response

HTTP response code and JSON data that represents the service configuration attributes. The parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/felb/configuration/services/{service_name}
```

Response:

```
200 ok

{
"active":"true/false",
"name":"service name",
"address":"ip_address",
"vip_nmask":"subnet mask",
"timeout":"integer",
"reentry":"integer",
"port":"integer",
```

```
"scheduler":"lc/rr/wlc/wrr/lblc/dh",
"persistent":"integer"

}
```

Retrieving a list of the servers for a service with web service

To retrieve a list of the current servers for a service, issue an HTTP GET command on the FELB service resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}/servers
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the service to retrieve a list of servers for

Response

HTTP response code and JSON data that represents the IP address of the server.

Parameter	Description
id	IP address of the server

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/felb/configuration/services/{service_name}/servers
```

Response:

```
200 ok
[
{"id":"Server address"},
{"id":"Server address"}
....
]
```

Retrieving server configuration attributes with web service

To retrieve a list of the current server configuration attributes with the RESTful web service, issue an HTTP GET command on the FELB server resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}
/servers/{address}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the service to retrieve server configuration attributes for
address	IP address of the server

Response

HTTP response code and JSON data that represents the server configuration attributes. The attribute parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/felb/configuration/services/{service_name}
/servers/{address}
```

Response:

```
200 ok
{
"active":"true/false",
"address":"ip_address",
"weight":"integer"
}
```

Creating a service with web service

To complete this operation with the RESTful web service, issue an HTTP POST command on the FELB service resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Table 9. Service level attributes

Parameter	Description
appliance_hostname	Host name of the appliance.
active	Specifies whether the new service is active. The value is `true` or `false`. By default, this value is set to `false`.
name	Specifies the name of the new service. This parameter is required.
address	Specifies the IP address that connects this service to the public network. This parameter is required.
vip_nmask	Specifies the network mask to be applied to the IP address for the service. This parameter is required.
timeout	Specifies the time in seconds that a service must be inactive before it is removed from the routing table. This parameter is required.
reentry	Specifies the time in seconds that a service must be active before it si added back into the routing table after it is previously removed because of failure. This parameter is required.
port	Specifies the port on which this service listens. This parameter is required.
scheduler	Specifies the scheduling algorithm for distributing jobs to the real servers. The choices are: **lc** Least connections **rr** Round robin **wlc** Weighted least connections **wrr** Weighted round robin **lblc** Locality based least connection This parameter is required.
persistent	Specifies the time (in seconds) to allow a persistent service connection to remain active. If this value is missing or set to 0, persistence is turned off. This parameter is required.
interface	Specifies the appliance interface on which the new service connects to the public network. The value must one of the application interface names (P.1, P.2, P.3, P.4). This parameter is required.
serverCount	The number of real servers that are attached to this service.

Table 10. Server level attributes

Parameter	Description
active	Specifies whether the new server is active. The value is `true` or `false`. By default, this value is set to `false`.
address	Specifies the IP address for the real server. This parameter is required.
weight	Specifies an integer that represents the processing capacity of this server relative to that of other real servers. For example, a server assigned 2000 has twice the capacity of a server assigned 1000. The weighted scheduling algorithms adjust this number dynamically based on workload. This parameter is required.

Response

HTTP response code and JSON data that represents the new service name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/felb/configuration/services

POST_DATA:
{
"active":"true/false",
"name":"service name",
"address":"ip_address",
"vip_nmask":"subnet mask",
"timeout":"integer",
"reentry":"integer",
"port":"integer",
"scheduler":"lc/rr/wlc/wrr/lblc/dh",
"persistent":"integer",
"interface":"interface id",

"servers":
            [
                {
                "active":"true/false",
                "address":"ip_address",
                "weight":"integer"
                },
                {
                ...
                }

            ]
}
```

Response:

```
200 ok

{"id":"The new service name"}
```

Creating a server with web service

To create a server with the RESTful web service, issue an HTTP POST command on the FELB server resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}/servers
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hosname	Host name of the appliance.
service_name	Name of the service to create a server for.
active	Specifies whether the new server is active. This parameter is required.
address	Specifies the IP address for the real server. This parameter is required.
weight	Specifies an integer that represents the processing capacity of this server relative to that of other real servers. For example, a server assigned 2000 has twice the capacity of a server assigned 1000. The weighted scheduling algorithms adjust this number dynamically based on workload. This parameter is required.

Response

HTTP response code and JSON data that represents the new server address.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/felb/configuration/services/{service_name}/servers

POST_DATA:
{
```

```
"active":"true/false",
"address":"ip_address",
"weight":"integer"
}
```

Response:

```
200 ok

{"id":"The new server address"}
```

Replacing front-end load balancer configuration in full with web service

To complete this operation with the RESTful web service, issue an HTTP PUT command on the FELB file resource URI.

URL

```
https://{appliance_hostname}/felb
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

The parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/felb

PUT_DATA: {
{
"enabled":"true/false",
"is_primary":"true/false",
"debug":"true/false",
"remote_active":"true/false",
"local":"ip address",
"remote":"ip address",
"keepalive":"integer",
"deadtime":"integer",
"nat_router":"ip address",
"nat_interface":"interface id",
"nat_nmask":"subnet mask",
"services":[{
            "active":"true/false",
```

```
            "name":"service name",
            "address":"ip address",
            "vip_nmask":"subnet mask",
            "timeout":"integer",
            "reentry":"integer",
            "port":"port number",
            "scheduler":"lc/rr/wlc/wrr/lblc/dh",
            "persistent":"integer",
            "interface":"interface id",
            "serverCount":"integer",
            "servers":[{
                       "active":"true/false",
                       "address":"ip address",
                       "weight":"integer"
                      }]
          }]
}
}
```

Response:

```
200 ok
```

Updating front-end load balancer level configuration attributes with web service

To complete this operation with the RESTful web service, issue an HTTP PUT command on the FELB file resource URI.

URL

```
https://{appliance_hostname}/felb/configuration
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

The parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/felb/configuration

PUT_DATA:
{
"enabled":"true/false",
```

```
"is_primary":"true/false",
"debug":"true/false",
"remote_active":"true/false",
"local":"ip address",
"remote":"ip address",
"keepalive":"integer",
"deadtime":"integer",
"nat_router":"ip address",
"nat_interface":"interface id",
"nat_nmask":"subnet mask"
}
```

Response:

```
200 ok
```

Updating service configuration attributes with web service

To update service configuration attributes with the web service, issue an HTTP PUT command on the FELB service resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the service to update configuration attributes for.

Other parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271. Any or all of these parameters can be supplied. This operation updates only the parameters that have been passed. The other existing parameters remain unchanged.

Note: The back-end address is set as the default gateway for load balanced servers. After you enable the front-end load balancer, ensure that you set the default gateway of each back-end server as the value set in the back-end `address` field.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/felb/configuration/services/{service_name}

PUT_DATA:
{
"active":"true/false",
"name":"service name",
"address":"ip_address",
"vip_nmask":"subnet mask",
"timeout":"integer",
"reentry":"integer",
"port":"integer",
"scheduler":"lc/rr/wlc/wrr/lblc/dh",
"persistent":"integer"
}
```

Response:

```
200 ok
```

Updating server configuration attributes with web service

To update the current server configuration attributes with the web service, issue an HTTP PUT command on the FELB server resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}
/servers/{address}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the service to update server configuration attributes for.
address	IP address of the server.

This command triggers an update to the server configuration by replacing the current parameters with the supplied parameters. These parameters are described in "Retrieving front-end load balancer configuration in full with web service" on page 271.

Note: Any or all of these parameters can be supplied. This operation updates only the parameters that have been passed. The other existing parameters remain unchanged.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/felb/configuration/services/{service_name}
/servers/{address}

PUT_DATA:
{
"active":"true/false",
"address":"ip_address",
"weight":"integer"
}
```

Response:

```
200 ok
```

Deleting a service with web service

To complete this operation with the RESTful web service, issue an HTTP DELETE command on the FELB service resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the service to delete.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/felb/configuration/services/{service_name}
```

Response:

```
200 ok
```

Deleting a server with web service

To delete a server with the RESTful web service, issue an HTTP DELETE command on the FELB server resource URI.

URL

```
https://{appliance_hostname}/felb/configuration/services/{service_name}
/servers/{address}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
service_name	Name of the server to delete
address	IP address of the server

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/felb/configuration/services/{service_name}
/servers/{address}
```

Response:

```
200 ok
```

Hosts file management

You can use the IBM Web Security Gateway Appliance Local Management Interface (LMI) to manage hosts file.

Managing hosts file with local management interface

To manage hosts file with the local management interface, use the Hosts File management page.

Procedure

1. From the top menu, select **Manage System Settings** > **Network Settings** > **Hosts File**. All current host records with their IP address and host names are displayed.
2. You can then work with host records and host names.
   - **Add a host record**
     a. Select the root level **Host Records** entry or do not select any entries.
     b. Click **New**.
     c. On the Create Host record page, provide IP address and host name of the host record to add.
     d. Click **Save**.
   - **Add a host name to a host record**
     a. Select the host record entry to add the host name to.
     b. Click **New**.
     c. On the Add Hostname to Host Record page, enter the host name to add.
     d. Click **Save**.
   - **Remove a host record**
     a. Select the host record entry to delete.
     b. Click **Delete**.
     c. On the confirmation page, click **Yes** to confirm the deletion.
   - **Remove a host name from a host record**
     a. Select host name entry to delete.
     b. Click **Delete**.
     c. On the confirmation page, click **Yes** to confirm the deletion.

     Note: If the removed host name is the only associated host name for the IP address, then the entire host record (the IP address and host name) is removed.

Retrieving the list of host IP addresses with web service

To retrieve the list of host IP addresses with the web service, issue an HTTP GET command on the host records resource URI.

URL

```
https://{appliance_hostname}/host_records
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the list of host IP addresses.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/host_records
```

Response:

```
200 ok

[
{"addr":"IP address 1"},
{"addr":"IP address 2"}
]
```

Retrieving the list of host names associated with a host IP address with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the host records resource URI.

URL

```
https://{appliance_hostname}/host_records/{host_address}/hostnames
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
host_address	The IP address to retrieve host names for.

Response

HTTP response code and JSON data that represents the list of host names for the specified host IP address.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/host_records/{host_address}/hostnames
```

Response:

```
200 ok

[
{
"name","server1"
},
{
"name","server2"
}
{
...
}
]
```

Adding a host name to a host IP address with web service

To add a host name to a host IP address with the web service, issue an HTTP POST command on the host records resource URI.

URL

```
https://{appliance_hostname}/host_records/{host_address}/hostnames
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
host_address	The IP address to add the host name.
name	The host name to add.

Response

HTTP response code and JSON data that represents the new host name.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/host_records/{host_address}/hostnames

POST_DATA:
{
"name":"The hostname to add"
}
```

Response:

```
200 ok

{"id":"The new hostname"}
```

Note: The hosts file is updated after the changes are deployed.

Creating a host record (IP address and host name) with web service

To create a host record (IP address and host name) with the web service, issue an HTTP POST command on the host records resource URI.

URL

```
https://{appliance_hostname}/host_records
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
addr	The IP address for the new host record.
hostnames	The list of host names (one or more) to associate with the IP address.
name	The name of a single host name to associate with the IP address.

Response

HTTP response code and JSON data that represents the new host.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/host_records

POST_DATA:
{
"addr":"The host IP address to create",
"hostnames":[
    {"name":"The hostname to associate"},
    {"name":"The hostname to associate"},
]
}
```

Response:

```
200 ok

{"id":"The new host record IP address"}
```

Removing a host record (IP address and associated host names) with web service

To remove a host record (IP address and associated host names) with the RESTful web service, issue an HTTP DELETE command on the host records resource URI.

URL

```
https://{appliance_hostname}/host_records/{host_address}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
host_address	The IP address of the host record to delete

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/host_records/{host_address}
```

Response:

```
200 ok
```

Note: The hosts file is updated after the changes are deployed.

Removing a host name from a host IP address with web service

To remove a host name from a host IP address with the web service, issue an HTTP DELETE command on the host records resource URI.

URL

```
https://{appliance_hostname}/host_records/{host_address}/hostnames/{hostname}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
host_address	The IP address of the host record to delete the host name from.
hostname	The host name to delete.

Note: If there are no further host names that are associated with the IP address, then the entire host record is deleted (that is, the IP address is removed).

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/host_records/{host_address}
/hostnames/{hostname}
```

Response:

```
200 ok
```

Packet tracing

You can manage packet tracing with either the local management interface or the web service.

Managing packet tracing with local management interface

To manage packet tracing with the local management interface, use the Packet Tracing management page.

Procedure

1. From the top menu, select **Manage System Settings** > **Network Settings** > **Packet Tracing**. The status of packet tracing is displayed.
2. Manage packet tracing settings.
   - **Start packet tracing**
     a. Click **Start**.
     b. On the Start Packet Tracing page:
        1) Select the interface name in the **Interface** field.

           Note: If no value is selected for the **Interface** field, packet tracing is enabled for all interfaces.
        2) Click the **Filter** field.
        3) On the Set Filter page, select a pre-defined filter in the **Display Filter** field, or enter the filter manually in the **Filter String** field.

4) Click **Save**.
5) Define the maximum size of the packet tracing file (PCAP file) in the **Maximum File Size** field. This value is the maximum size that the packet tracing file can grow to before packet tracing is disabled.

   Note: If no value is selected for the **Maximum File Size** field, the maximum file size is set to half the remaining disk size.

c. Click **Start**.

   Note: Only a single packet tracing operation can be running at the same time. A new packet trace cannot be started until the PCAP file from the previous trace is deleted.

- **Stop packet tracing**
  a. Click **Stop**.
  b. Click **Yes** to confirm the action.
- **Export the packet tracing PCAP file**
  a. Click **Export**.

     Note: You must configure the software that blocks pop-up windows in your browser to allow pop-up windows for the appliance before files can be exported.
  b. Confirm the save action in the browser pop-up window.
- **Delete the packet tracing PCAP file**
  a. Click **Delete**.
  b. Click **Yes** to confirm the action.

     Note: If packet tracing is running, the PCAP file cannot be deleted. You must stop the associated packet tracing before you delete the PCAP file.

Starting packet tracing with web service

To start packet tracing with the RESTful web service, issue an HTTP PUT command on the packet tracing resource URI.

URL

```
https://{appliance_hostname}/packet_tracing
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Name	Description
appliance_hostname	Host name of the appliance.
enable	Indicates that packet tracing is started. This parameter is required.
filter	The filter rule for the tracing, conforming to the PCAP filtering syntax. This parameter is required.

Name	Description
interface	The name of the interface to enable packet tracing on. The value is `P.1`, `P.2`, `P.3`, `P.4`, `M.1`, or `M.2`. **Note:** If no value is specified for this parameter, packet tracing is enabled for all interfaces.
max_size	The maximum size (in MB) of the output trace file. The value must be larger than 0. If not specified, the maximum size is set to half the remaining disk size.

Response

HTTP response code and JSON data that represents the packet tracing settings.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/packet_tracing

PUT_DATA:
{
"enable":"true",
"filter":"packet tracing filter",
"interface":"interface name (eg: P.1)",
"max_size":"maximum PCAP file size (in MB)"
}
```

Response:

```
200 ok

{
"enable":"true",
"filter":"packet tracing filter",
"interface":"interface name (eg: P.1)",
"max_size":"maximum PCAP file size (in MB)"
}
```

Notes:

- Packet capturing cannot run in promiscuous mode. Only packets that are destined for the configured interface are captured.
- Packet capturing can be started only after the existing PCAP file is cleared. If the PCAP file is not 0 bytes, then perform the clear operation first. Otherwise, this operation results in an error. For details about how to clear the existing PCAP file, see "Clearing the packet tracing PCAP file with web service" on page 298.
- Only a single packet trace operation can be running at any time

Stopping packet tracing with web service

To stop packet tracing with the RESTful web service, issue an HTTP PUT command on the packet tracing resource URI.

URL

```
https://{appliance_hostname}/packet_tracing
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
enable	Indicates that packet tracing is stopped. This parameter is required. The value is `false` in this case.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
PUT https://{appliance_hostname}/packet_tracing

PUT_DATA:
{
"enable":"false"
}
```

Response:

```
200 ok
```

Viewing the packet tracing status with web service

To view the packet tracing status with the web service, issue an HTTP GET command on the packet tracing resource URI.

URL

```
https://{appliance_hostname}/packet_tracing
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents a short version of the status of packet tracing.

Parameter	Description
enabled	A boolean that indicates whether packet tracing is enabled.
status	An array of name-value pairs that represents the status.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/packet_tracing

{
"enabled":"true/false",
"status":"The running status"
}
```

Response:

```
200 ok

{
"enabled":"true",
"status":[
{"id":"State","value":"running"},
{"id":"Interface","value:"P.1"},
{"id":"Filter Rule","value":"dst port 80"},
{"id":"Max File Size (bytes)","value":"1240000"},
{"id":"File Size (bytes)","value":"1112345"},
{"id":"Packet Count","value":"212"},
{"id":"Time of First Packet","value":"Fri Jun 29 14:14:02 2012"},
{"id":"Time of Last Packet","value":"Fri Jun 29 14:14:17 2012"},
{"id":"Average Packet Size (bytes)","value":"205.36"},
{"id":"Average Packet Rate (packets/sec)","value":"42.20"}]
}
```

Exporting the packet tracing PCAP file with web service

To export the packet tracing PCAP file with the web service, issue an HTTP GET command on the packet tracing resource URI.

URL

```
https://{appliance_hostname}/packet_tracing/pcap?export
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and the PCAP file.

The PCAP file must be viewed in an external viewer. Wireshark is an example of one such viewer.

Notes:

- For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.
- The output of the web service must be redirected to a local file as it is not screen readable.

Example

Request:

```
GET https://{appliance_hostname}/packet_tracing/pcap?export
```

Response:

```
200 ok

The PCAP file
```

Clearing the packet tracing PCAP file with web service

To clear the packet tracing PCAP file with the web service, issue an HTTP DELETE command on the packet tracing resource URI.

URL

```
https://{appliance_hostname}/packet_tracing
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
DELETE
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
DELETE https://{appliance_hostname}/packet_tracing
```

Response:

```
200 ok
```

System settings

Information about managing system settings on your Security Web Gateway Appliance.

Configuring date and time settings

Use the Date/Time Configuration page to configure the date, time, time zone, and NTP server information.

Procedure

1. Click **Manage System Settings** > **System Settings** > **Date/Time**
2. Configure the following options:

Option	Description
Time Zone	Specifies the time zone for the appliance.
Date/Time	Specifies the day, month, year, and time for the appliance.
NTP Server address	Lists the NTP (NIST Internet Time Service) servers the appliance uses. You can enter multiple NTP servers, separated by commas.

3. Click **Save**.

Configuring administrator settings

Use administrator settings to change the password you use to access the appliance and the length of idle time that is allowed to pass before your session times out.

Procedure

1. Click **Manage System Settings** > **System Settings** > **Administrator Settings**.
2. On the Administrator Settings page, type your current password in the **Current Password** field.
3. Type your new password in the **New Password** field.
4. Type your new password in the **New Password Confirmation** field.
5. In the **Session Timeout** field, click the arrows to select the amount of time you are allowed to be idle before you are automatically logged out.
6. Click **Save**.

Management authentication

Use either the local management interface or web service to manage LMI authentication methods.

Configuring management authentication with local management interface

To configure management authentication with the local management interface, use the Management Authentication management page.

Procedure

1. From the top menu, select **Manage System Settings** > **System Settings** > **Management Authentication**. All current management authentication settings are displayed.
2. In the Main tab:
   - Select **Local User Database** if you want to use the local user database for authentication.
   - Select **Remote LDAP User Registry** if you want to use the remote LDAP user registry for authentication.
     a. In the LDAP tab:
        1) Specify the name of the LDAP server in the **Host name** field.
        2) Specify the port over which to communicate with the LDAP server in the **Port** field.
        3) Select the **Anonymous Bind** check box if the LDAP user registry supports anonymous bind.
        4) Specify the DN of the user that is used to bind to the registry in the **Bind DN** field.
        5) Specifies the password that is associated with the bind DN in the **Bind Password** field.
     b. In the LDAP General tab:
        1) Specify the name of the LDAP attribute that holds the supplied authentication user name of the user in the **User Attribute** field.
        2) Specify the name of the LDAP attribute that is used to hold the members of a group in the **Group Member Attribute** field.
        3) Specify the base DN that is used to house all administrative users in the **Base DN** field.
        4) Specify the DN of the group to which all administrative users belong in the **Administrative Group DN** field.
     c. In the LDAP SSL tab:
        1) Select the **Enable SSL** check box to define whether SSL is used when the system communicates with the LDAP server.
        2) Select the name of the key database file in the **Key File Name** field.
        3) Select the name of the certificate to be used if client authentication is requested by the LDAP server in the **Certificate Label** field.
3. Click **Save** to save your settings.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.
4. *Optional*: Click **Test** to test the authentication.

   Note: If there have been changes made to the management authentication configuration that have not yet been deployed, this test will run using the undeployed configuration.
   a. In the Test Authentication window, enter the user name in the **Username** field.
   b. Enter the password in the **Password** field.
   c. Click **Test**.

   If the authentication is successful, a success message is displayed. If the authentication is not successful, an error message is displayed.

Retrieving the current authentication configuration with web service

To complete this operation with the RESTful web service, issue an HTTP GET command on the management authentication resource URI.

URL

```
https://{appliance_hostname}/management_authentication
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the configuration.

Table 11. Authentication configuration parameters

Field	Description
type	Specifies whether the local user database or the remote LDAP user registry is used for authentication. If this parameter is set to `local`, then all other fields are ignored. The value is `local` or `remote`.
ldap_host	Specifies the name of the LDAP server. This parameter is required if type is `remote`.
ldap_port	Specifies the port over which to communicate with the LDAP server. This parameter is required if type is remote.
enable_ssl	Specifies whether SSL is used when the system communicates with the LDAP server. The value is `true` or `false`.
key_database	Specifies the name of the key database file (without any path information). This parameter is required if **enable_ssl** is set to `true` and type is `remote`.
cert_label	Specifies the name of the certificate within the Key database that is used if client authentication is requested by the LDAP server. This parameter is optional.
user_attribute	Specifies the name of the LDAP attribute which holds the supplied authentication user name of the user. This parameter is required if type is `remote`.
group_member_attribute	Specifies the name of the LDAP attribute which is used to hold the members of a group. This parameter is required if type is `remote`.
base_dn	Specifies the base DN which is used to house all administrative users. This parameter is optional.

Table 11. Authentication configuration parameters (continued)

Field	Description
admin_group_dn	Specifies the DN of the group to which all administrative users must belong.
anon_bind	Specifies whether the LDAP user registry supports anonymous bind. If set to `false`, **bind_dn** and **bind_password** are required. The value is `true` or `false`.
bind_dn	Specifies the DN of the user which will be used to bind to the registry. This user must have `read` access to the directory. This parameter is required if **anon_bind** is `false` and type is `remote`.
bind_password	Specifies the password which is associated with the **bind_dn**. This parameter is required if **anon_bind** is `false` and type is `remote`.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/management_authentication
```

Response:

```
200 ok

{
"type":"local/remote",
"ldap_host":"ldap server",
"ldap_port":"ldap port",
"enable_ssl":"true/false",
"key_database":"key database",
"cert_label":"certificate label",
"user_attribute":"user attribute",
"group_member_attribute":"group member attribute",
"base_dn":"base DN",
"admin_group_dn":"administrative group DN",
"anon_bind":"true/false",
"bind_dn":"bind DN"
}
```

Updating the current authentication configuration with web service

To update (replace) the current authentication configuration with the RESTful web service, issue an HTTP PUT command on the management authentication resource URI.

URL

```
https://{appliance_hostname}/application_interfaces/{interface_id}/addresses/{id}
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
PUT
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

The parameters are described in Table 11 on page 301.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

To set as local user database:

```
PUT https://{appliance_hostname}/management_authentication

PUT_DATA:
{
"type":"local"
}
```

To set as remote LDAP user registry:

```
PUT https://{appliance_hostname}/management_authentication

PUT_DATA:
{
"type":"remote",
"ldap_host":"ldap server",
"ldap_port":"ldap port",
"enable_ssl":"true/false",
"key_database":"key database",
"cert_label":"certificate label",
"user_attribute":"user attribute",
"group_member_attribute":"group member attribute",
"base_dn":"base DN",
"admin_group_dn":"administrative group DN",
"anon_bind":"true/false",
"bind_dn":"bind DN",
"bind_password":"bind password"
}
```

Response:

```
200 ok

{
"type":"local/remote",
"ldap_host":"ldap server",
"ldap_port":"ldap port",
"enable_ssl":"true/false",
"key_database":"key database",
"cert_label":"certificate label",
"user_attribute":"user attribute",
"group_member_attribute":"group member attribute",
```

```
"base_dn":"base DN",
"admin_group_dn":"administrative group DN",
"anon_bind":"true/false",
"bind_dn":"bind DN"
}
```

Testing the management authentication with web service

To test the current management authentication settings with the RESTful web service, issue an HTTP POST command on the management authentication resource URI.

URL

```
https://{appliance_hostname}/management_authentication
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.
user	Name of the user to test authentication with.
password	Password of the user to test authentication with.

Response

HTTP response code and JSON error message where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/management_authentication

POST_DATA:
{
"user":"The name of the user to test",
"password":"The password of the user to test"
}
```

Response:

```
200 ok
```

Management SSL certificate

Use either the local management interface or web service to work with the management SSL certificate.

Note: If the certificate expires, the local management interface is not reachable. In this situation, use the `reset_lmi_cert` CLI command within the LMI menu to

generate a self-signed certificate so that access to the LMI can be re-established. Then, use the `restart` CLI command to restart the local management interface.

You can type `help` in the command-line interface for a list of commands available in the current mode. For example:

```
webapp.vwasp.gc.au.ibm.com:lmi>help
Current mode commands:
reset_lmi_cert  Reset the server certificate for the local management interface
                to a self-signed certificate.
restart         Restart the local management interface.
```

Working with management SSL certificate with local management interface

In the local management interface, go to **Manage System Settings** > **System Settings** > **Management SSL Certificate**.

Viewing the details of the current management SSL certificate with local management interface:

To view the details of the current management SSL certificate with the local management interface, use the Management SSL Certificate page.

Procedure

1. From the top menu, select **Manage System Settings** > **System Settings** > **Management SSL Certificate**.
2. The details of the current management certificate are displayed.

Updating the management SSL certificate with local management interface:

To update the management SSL certificate with the local management interface, use the Management SSL Certificate page.

Procedure

1. From the top menu, select **Manage System Settings** > **System Settings** > **Management SSL Certificate**.
2. Select **Update**.
3. Under **Certificate File**, click **Browse**.
4. Browse to the directory that contains the certificate container file and select the file.

   Note: The certificate container file must be PKCS12 format (.p12 file) and can contain only a single certificate. This certificate is used as the management SSL certificate.
5. Click **Open**.
6. Click **Update**. A message that indicates successful update is displayed.

   Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Working with management SSL certificate with web service

Use the management SSL certificate resource URI.

Retrieving the details of the current management SSL certificate with web service:

To complete the operation with the RESTful web service, issue an HTTP GET command on the management SSL certificate resource URI.

URL

```
https://{appliance_hostname}/management_ssl_certificate
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
GET
```

Parameters

Parameter	Description
appliance_hostname	Host name of the appliance.

Response

HTTP response code and JSON data that represents the details of the management SSL certificate.

Parameter	Description
keysize	The key size of the certificate
version	The version of the certificate
issuer	The issuer of the certificate
subject	The subject of the certificate
notbefore	Start date of the certificate validity
notafter	Expiry date of the certificate validity

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
GET https://{appliance_hostname}/management_ssl_certificate
```

Response:

```
200 ok

{
"keysize":"The certificate key size",
"version":"cert_version",
"issuer":"cert_issuer",
"suject":"cert_subject",
"notbefore":"date",
"notafter":"date"
}
```

Updating the management SSL certificate with web service:

To update the management SSL certificate with the RESTful web service, issue an HTTP POST command on the management SSL certificate resource URI.

URL

```
https://{appliance_hostname}/management_ssl_certificate
```

Note: The web service request must use the format that is described in "Required header for calling a web service" on page 10.

Method

```
POST
```

Note: For the changes to take effect, they must be deployed as described in "Configuration changes commit process" on page 28.

Parameters

Description	Description
appliance_hostname	Host name of the appliance.
cert	Name of the file that contains the certificate to import. This parameter is required. **Note:** The certificate container file must be PKCS12 format (`.p12` file) and can contain only a single certificate. This certificate is used as the management SSL certificate.
password	The password of the source certificate container. This parameter is optional.

Response

HTTP response code and JSON error response where applicable.

Note: For descriptions of possible error responses that can be returned from a web service call, see "Web service responses" on page 11.

Example

Request:

```
POST https://{appliance_hostname}/management_ssl_certificate

POST_DATA: {
"cert":"The certificate to import" (as a file),
"password":"The password of the source certificate container"
}
```

Response:

```
200 ok
```

Managing advanced tuning parameters

Change advanced tuning parameter values only under the supervision of IBM software support.

Managing snapshots

Use snapshots to restore prior configuration and policy settings to the Security Web Gateway Appliance. It is recommended to back up the appliance on a frequent basis by downloading snapshot files.

About this task

Snapshots are stored on the appliance. However, you can download snapshots to an external drive in case of system failure.

Procedure

1. Click **Manage System Settings** > **System Settings** > **Snapshots**.
2. In the Snapshots pane, use one or more of the following commands:

Option	Description
New	To create a snapshot, click **New**, type a comment that describes the snapshot, and then click **Save**.
Edit	To edit the comment for a snapshot, select the snapshot, click **Edit**, type a new comment, and then click **Save**.
Delete	To delete snapshots, select one or more snapshots, and then click **Delete**.
Apply	To apply a snapshot, select the snapshot, and then click **Apply**. **Note:** If configuration or policy versions are newer than the firmware version, the settings are rejected. If the configuration and policy versions are older than the firmware version, the settings are migrated to the current firmware version.
Download	To download a snapshot, select the snapshot, click **Download**, browse to the drive where you want to save the snapshot, and then click **Save**. **Note:** If you download multiple snapshots, the snapshots are compressed into a .zip file.
Upload	To upload snapshots, click **Upload**, browse to the snapshots you want to upload, select the snapshots, and then click **Save**. **Note:** You can upload only one snapshot at a time.
Refresh	To refresh the list of snapshots, click **Refresh**.

Managing support files

IBM Customer Support uses support files to help you troubleshoot problems with the appliance. Support files contain all log files, temporary and intermediate files, and command output that is needed to diagnose customer support problems.

About this task

Support files might contain customer-identifiable information, such as IP addresses, host names, user names, and policy files. Support files might also contain confidential information, such as passwords, certificates, and keys. The support file contents are stored as a .zip file. All files inside the support file can be inspected and censored by the customer.

Tip: You can create multiple support files to track an issue over time.

Procedure

1. Click **Manage System Settings** > **System Settings** > **Support Files**.
2. In the Support Files pane, use one or more of the following commands:

Option	Description
New	To create a support file, click **New**, type a comment that describes the support file, and then click **Save**. A new support file is created on the appliance.
Edit	To edit the comment for a support file, select the support file, click **Edit**, type a new comment, and then click **Save**.
Delete	To delete a support file, select the support file, and then click **Delete**.
Download	To download support files, select the support files, click **Download**, browse to the drive where you want to save the support files, and then click **Save**. **Note:** If you download multiple support files, the files are compressed into a .zip file.

Configuring system alerts

Configure where you want the system to send notifications about changes to system settings and problems with the system.

About this task

Available alerts include system alerts pre-defined in the system and any alert objects that you created.

Procedure

1. Click **Manage System Settings** > **System Settings** > **System Alerts**.
2. In the System Alerts pane, complete one or more of the following tasks:
   - To receive notifications for problems with the system, select one or more system alert objects from the Available Objects pane, and add them.
   - To create or edit alert objects, see these related topics to configure one or more of the following alert objects:
     - "Configuring email alert objects" on page 310
     - "Configuring remote syslog alert objects" on page 311
     - "Configuring SNMP alert objects" on page 310
   - To delete a system alert, select the alert and then click **Delete**.

Configuring SNMP alert objects

Configure SNMP alert objects to enable the system to send system alerts to an SNMP Manager.

Procedure

1. Click **Manage System Settings** > **System Settings** > **System Alerts**.
2. In the System Alerts page, take one of the following actions:
   - Click **New** > **SNMP**.
   - Select an existing object, and then click **Edit**.
3. Type a name for the alert object.
4. Select a trap version from the list.
5. In the SNMP Manager box, type the IP address, host name, or fully qualified domain name (FQDN) of the SNMP manager.

   Note: The SNMP host must be accessible to the appliance to send SNMP traps.
6. Type the port number that the SNMP manager monitors for notifications.

   Note: The default port number is 162.
7. Type a comment to describe the SNMP alert object.
8. For trap versions V1 or V2c, type the name of the community that is used to authenticate with the SNMP agent.
9. For trap version 3, configure the following options:

Option	Description
Name	Type the user name to be authenticated in the SNMP database.
Notification Type	On the **Notification Type** tab, select **Inform** or **Trap** in the **SNMP Trap Version** field.
Authentication	On the **Authentication and Privacy** tab, select **Enabled** to enable authentication, type the authentication passphrase, and then select an authentication type.
Privacy	Select **Enabled** to enable privacy, type the privacy passphrase, and then select a privacy type.

10. Click **Save**.

Configuring email alert objects

You can create email alert objects to send an email notification to specified users or to administrators when specified events occur on your network. You can also select the event parameters to include in the message so that important information about detected events is provided.

Procedure

1. Click **Manage System Settings** > **System Settings** > **System Alerts**.
2. In System Alerts page, take one of the following actions:
   - Click **New** > **Email**.
   - Select an existing object, and then click **Edit**.
3. Configure the following options:

Option	Description
Name	Specifies a meaningful name for the response. **Note:** This name displays when you select responses for events, so give the response a name that allows users to easily identify what they are selecting.
From	Specifies the email address that displays in the **From** field of the alert email.
To	Specifies the email address or group of addresses to receive the alert. **Note:** Separate individual email addresses with a comma or semicolon.
SMTP Server	Specifies the fully qualified domain name or IP address of the mail server. **Note:** The SMTP server must be accessible to the appliance to send email notifications.
SMTP Port	Specifies the custom port that is used to connect to the SMTP server. The default is 25.
Comment	Type a comment to identify the email alert object.

4. Click **Save**.

Configuring remote syslog alert objects

Configure remote syslog alert objects to enable the system to record system events in a remote log file.

Procedure

1. Click **Manage System Settings** > **System Settings** > **System Alerts**.
2. In the System Alerts page, do one of the following steps:
   - Click **New** > **Remote Syslog**.
   - Select an existing remote syslog alert object, and then click **Edit**.
3. Configure the following options:

Option	Description
Name	Specifies a meaningful name for the response.
Remote Syslog Collector	Specifies the fully qualified domain name or IP address of the host on which you want to save the log. **Note:** The host must be accessible to the appliance.
Remote Syslog Collector Port	Specifies the custom port that is used to connect to the syslog collector. The default is 514.
Comment	Type a comment to identify the remote syslog alert object.

4. Click **Save**.

Restarting or shutting down the appliance

Use the Restart or Shut down page to restart or shut down the appliance.

About this task

Important: When the appliance is restarting or shutting down, traffic is not passed through the appliance and your network might not be protected.

Procedure

1. Click **Manage System Settings** > **System Settings** > **Restart or Shut down**
2. Perform one of the following tasks:

Option	Description
Click Restart to restart the appliance	Restarting the appliance takes it offline for several minutes.
Click Shut down to turn off the appliance	Shutting down the appliance takes it offline and makes it inaccessible over the network until you restart it.

3. Click **Yes**.

Chapter 8. Troubleshooting

Information about how to troubleshoot a problem with the appliance.

IPMItool

If you are using the hardware appliance, you can use the IPMItool to manage the Baseboard Management Controller (BMC) module. The IPMItool cannot be used with the virtual appliance.

The IPMItool is a utility to monitor, configure, and manage devices that support the Intelligent Platform Management Interface (IPMI). IPMI is a standardized message-based hardware management interface. A hardware chip that is known as the Baseboard Management Controller (BMC), or Management Controller (MC), implements the core of IPMI.

The BMC provides key interfaces that are needed for monitoring the health of the system hardware. There are interfaces for user channels, monitoring elements (temperature, voltage, fan speed, bus errors, and so on), manually driven recovery (local or remote system resets and power on/off operations), and an interface for logging without operating system intervention for abnormal or 'out-of-range' conditions for later examination and alerting. It is important to note that the BMC is always powered ON. It contains a small processor that runs IPMI even when the main system is OFF, or the operating system has crashed. So the BMC can be configured to look at the status of local hardware from another server for secure remote monitoring and recovery (such as system reset) regardless of the status of the platform.

To access the IPMItool, follow these steps:

1. Log on to the command-line interface (CLI) as a root user.
2. Enter the `Hardware` menu.
3. Enter `ipmitool`.

   Note: The IPMItool can be run only with the hardware appliance. Attempts to enter the IPMItool from a virtual appliance result in errors that are returned.
4. Enter specific commands to manage the BMC module as needed.

   To see a list of IPMItool commands, enter `# ipmitool help`.

   You can also get help for many specific IPMItool commands by adding the word help after the command. For example, `# ipmitool channel help`.

Running self-diagnostic tests (hardware appliance only)

The hardware appliance provides a self-diagnostic program to assist with troubleshooting. This feature is not available with the virtual appliance.

Procedure

1. Reboot the appliance.
2. From the console, select the **Hardware Diagnostics** option from the GNU GRUB menu. The diagnostics program starts running after this option is selected. After the appliance finishes booting, the diagnostics program is accessible from the console.

3. To run a diagnostic test, enter `phdiag <test_name>`, where `<test_name>` is one of the following values in bold:

   all Run all standard tests (No storage bad blocks test, default)

   lcd Run LCD tests

   system
   Run standard system tests (MTM-Serial, Inventory, PSU, FAN, SEL)

   network
   Run network port tests (Selftest, Traffic)

   storage
   Run storage tests (SMART, FSCK)

   badblocks
   Run storage bad blocks test

Error HPDBG1005E: Could not contact the LDAP server

This error occurs during the configuration of the runtime environment on a remote system against a stand-alone Policy Server and LDAP on the appliance.

A separate LDAP server is required to configure the runtime environment on Windows against a stand-alone Policy Server and LDAP on the appliance. You must set up this LDAP server before you start the runtime environment configuration. It can either be an existing LDAP server or a temporary LDAP server set up just for this purpose. The LDAP is not modified during the runtime environment configuration.

This setup is required because the runtime environment on Windows checks whether an LDAP server is running during the initial configuration process. If no running LDAP server is detected, the runtime environment configuration fails. The runtime environment on Windows does not use this LDAP server other than the initial contact to check that it is running.

Installing firmware from a USB boot drive: Windows

Create a USB boot drive in a Windows OS and use it to install firmware on the Security Web Gateway Appliance.

About this task

You might choose to install new firmware to resolve software and configuration errors that cause your appliance not to work properly. For example, you can use this procedure to install new firmware on an appliance that does not boot due to a software error.

Refer to the IBM Support Portal for help troubleshooting appliance problems.

Procedure

1. Download the appliance firmware and save it to a secure host in your network.
2. Insert the USB flash drive into a USB port on the same host and note where the operating system assigns the USB flash drive.
3. Use an image writer program to overwrite the contents of the USB flash drive with the firmware image.

Tip: Common writer programs for Windows include Win32DiskImager.exe and USB Image Tool.

4. Turn off the appliance.
5. Connect the USB flash drive to the appliance and turn the appliance on. The appliance boots from the USB boot drive.
6. Log on to the installer command-line interface as an administrator.
   - `install login: admin`
   - `password:admin`

   Tip: You can type `help` for a list of commands available in the current mode.
7. Type `restore`.
8. Type `YES` and press Enter.

   Note: The installation takes approximately 30 minutes to complete. The firmware is installed and the appliance restarts.

Installing firmware from a USB boot drive: Linux

Create a USB boot drive in a Linux operating system and use it to install firmware on the appliance.

About this task

You might choose to install new firmware to resolve software and configuration errors that cause your appliance not to work properly. For example, you can use this procedure to install new firmware on an appliance that does not boot because of a software error.

See the IBM Support Portal for help troubleshooting appliance problems.

Procedure

1. Download the appliance firmware and save it to a secure host in your network.
2. Insert the USB flash drive into a USB port on the same host and note where the operating system assigns the USB flash drive.
3. Use an image writer program to overwrite the contents of the USB flash drive with the firmware image.

   Tip: USB ImageWriter is included in most Linux distributions.
4. Turn off the appliance.
5. Connect the USB flash drive to the appliance and turn on the appliance. The appliance boots from the USB boot drive.
6. Log on to the installer command-line interface as an administrator.
   - `install login: admin`
   - `password:admin`

   Tip: You can type `help` for a list of commands available in the current mode.
7. Type `restore`.
8. Type `YES` and press Enter.

   Note: The installation takes approximately 30 minutes to complete. The firmware is installed and the appliance restarts.

Installing firmware from a USB boot drive: Mac OS

Create a USB boot drive in a Mac OS and use it to install firmware on the appliance.

About this task

You might choose to install new firmware to resolve software and configuration errors that cause your appliance not to work properly. For example, you can use this procedure to install new firmware on an appliance that does not boot because of a software error.

Refer to the IBM Support Portal for help troubleshooting appliance problems.

Procedure

1. Download the appliance firmware and save it to a secure host in your network.
2. On the secure host, open the Terminal application.
3. In the Terminal application window, run `diskutil list` to get a current list of devices.
4. Connect the USB flash drive to the secure host.
5. Run `diskutil list` again and determine which device node the system assigned the USB device to.
6. Run `sudo dd if=/path/to/downloaded.img of=/dev/rdiskN bs=1m`. Replace `/path/to/downloaded.img` with the path to the firmware file.

   Note: If you get the following error, replace `bs=1m` with `bs=1M`:

   ```
   dd: Invalid number `1m', you are using GNU dd
   ```
7. Run `diskutil eject /dev/diskN` and remove your device after the command is complete.
8. Turn off the appliance.
9. Connect the USB flash drive to the appliance and turn on the appliance. The appliance boots from the USB boot drive.
10. Log on to the installer command-line interface as an administrator.
    - `install login: admin`
    - `password:admin`

    Tip: You can type `help` for a list of commands available in the current mode.
11. Type `restore`.
12. Type `YES` and press Enter.

    Note: The installation takes approximately 30 minutes to complete. The firmware is installed and the appliance restarts.

Erasing the hardware appliance: Windows

Perform a secure erasure on the hardware appliance when you want to make it impossible to recover any data that was previously on the drive.

About this task

You might erase an appliance as part of the process to return the hardware appliance to IBM for a replacement unit or before you discard the appliance. You can erase an appliance that does not boot because of software or configuration errors. You can also erase an appliance with some hardware failures. However, an erasure is unlikely to work on some hardware failures, such as a failed hard disk.

Erasing an appliance does not restore functionality to the appliance.

Procedure

1. Download the appliance firmware and save it to a secure host in your network.
2. Insert the USB flash drive into a USB port on the same host and note where the operating system assigns the USB flash drive.
3. Use an image writer program to overwrite the contents of the USB flash drive with the firmware image.

   Tip: Common writer programs for Windows include Win32DiskImager.exe and USB Image Tool.
4. Turn off the appliance.
5. Connect the USB flash drive to the appliance and turn on the appliance. The appliance boots from the USB boot drive.
6. Log on to the installer command-line interface as an administrator.
   - `install login: admin`
   - `password:admin`

   Tip: You can type `help` for a list of commands available in the current mode.
7. Type `wipe` and press Enter.

   Note: The erasure procedure takes approximately 30 minutes to complete.

Erasing the hardware appliance: Linux

Perform a secure erasure on the hardware appliance when you want to make it impossible to recover any data that was previously on the drive.

About this task

You might erase an appliance as part of the process to return the hardware appliance to IBM for a replacement unit or before you discard the appliance. You can erase an appliance that does not boot because of software or configuration errors. You can also erase an appliance with some hardware failures. However, an erasure is unlikely to work on some hardware failures, such as a failed hard disk.

Erasing an appliance does not restore functionality to the appliance.

Procedure

1. Download the appliance firmware and save it to a secure host in your network.
2. Insert the USB flash drive into a USB port on the same host and note where the operating system assigns the USB flash drive.
3. Use an image writer program to overwrite the contents of the USB flash drive with the firmware image.

Tip: USB ImageWriter is included in most Linux distributions.

4. Turn off the appliance.
5. Connect the USB flash drive to the appliance and turn on the appliance. The appliance boots from the USB boot drive.
6. Log on to the installer command-line interface as an administrator.
   - `install login: admin`
   - `password:admin`

   Tip: You can type `help` for a list of commands available in the current mode.
7. Type `wipe` and press Enter.

   Note: The erasure procedure takes approximately 30 minutes to complete.

Erasing the hardware appliance: Mac OS

Perform a secure erasure on the hardware appliance when you want to make it impossible to recover any data that was previously on the drive.

About this task

You might erase an appliance as part of the process to return the hardware appliance to IBM for a replacement unit or before you discard the appliance. You can erase an appliance that does not boot because of software or configuration errors. You can also erase an appliance with some hardware failures. However, an erasure is unlikely to work on some hardware failures, such as a failed hard disk.

Erasing an appliance does not restore functionality to the appliance.

Procedure

1. Download the appliance firmware and save it to a secure host in your network.
2. On the secure host, open the Terminal application.
3. In the Terminal application window, run `diskutil list` to get a current list of devices.
4. Connect the USB flash drive to the secure host.
5. Run `diskutil list` again and determine which device node the system assigned the USB device to.
6. Run `sudo dd if=/path/to/downloaded.img of=/dev/rdiskN bs=1m`. Replace `/path/to/downloaded.img` with the path to the firmware file.

   Note: If you get the following error, replace `bs=1m` with `bs=1M`:

   ```
   dd: Invalid number `1m', you are using GNU dd
   ```
7. Run `diskutil eject /dev/diskN` and remove your device after the command is complete.
8. Turn off the appliance.
9. Connect the USB flash drive to the appliance and turn on the appliance. The appliance boots from the USB boot drive.
10. Log on to the installer command-line interface as an administrator.
    - `install login: admin`
    - `password:admin`

Tip: You can type `help` for a list of commands available in the current mode.

11. Type `wipe` and press Enter.

    Note: The erasure procedure takes approximately 30 minutes to complete.

Technical support

IBM Security provides technical support to customers who are entitled to receive support.

The IBM Support Portal

Before you contact IBM Security Solutions about a problem, see the IBM Support Portal at http://www.ibm.com/software/support.

The IBM Software Support Guide

If you need to contact technical support, use the methods described in the IBM Software Support Guide at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html.

The guide provides the following information:

- Registration and eligibility requirements for receiving support
- Customer support telephone numbers for the country in which you are located
- Information you must gather before you call

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine and Cell/B.E. are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Index

G

H

I

J

K

L

U

V

W

X

Printed in USA

SC22-5432-00