Security Access Manager
Version 7.0

Shared Session Management Administration Guide

SC23-6509-02

Security Access Manager
Version 7.0

Shared Session Management Administration Guide

IBM

SC23-6509-02

Note

Before using this information and the product it supports, read the information in “Notices” on page 89.

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

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 session management server (SMS) manages sessions across clustered Security Access Manager security servers. Implemented as a WebSphere® Application Server service, the session management server permits the sharing of session information and provides a user interface from which authorized persons can administer and monitor user sessions.

For details about supported platforms, disk and memory requirements, see the *IBM Security Access Manager for Web: Release Notes*.

For details about software prerequisites and installation and initial configuration of the session management server components, see the *IBM Security Access Manager for Web: Installation Guide*.

For technical reference information, deployment considerations, and usage information for the session management server with Security Access Manager Plug-in for Web Servers, see the *IBM Security Access Manager for Web: Plug-in for Web Servers Administration Guide*.

For technical reference information, deployment considerations, and usage information for the session management server with Security Access Manager WebSEAL, see the *IBM Security Access Manager for Web: WebSEAL Administration Guide.*

Intended audience

This guide is for system administrators responsible for the deployment and administration of the Security Access Manager session management server.

Readers should be familiar with the following:

- Microsoft Windows and UNIX operating systems.
- Database architecture and concepts.
- Security management.
- Internet protocols, including HTTP, HTTPS and TCP/IP.
- 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."
- Links to "Online publications" on page xii.
- 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*, SC23-6504-03

  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*, SC23-6505-03

  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*, SC22-5432-01

  Provides administrative procedures and technical reference information for the WebSEAL Appliance.
- *IBM Security Web Gateway Appliance Configuration Guide for Web Reverse Proxy*, SC22-5433-01

  Provides configuration procedures and technical reference information for the WebSEAL Appliance.
- *IBM Security Web Gateway Appliance Web Reverse Proxy Stanza Reference*, SC27-4442-01

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*, SC27-4443-01

  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*, SC23-6511-03

  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*, SC23-6512-03

  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 Security Systems Documentation Central and Welcome page
IBM Security Systems Documentation Central provides an alphabetical list of all IBM Security Systems product documentation and links to the product information center for specific versions of each product.

Welcome to IBM Security Systems Information Centers provides and introduction to, links to, and general information about IBM Security Systems information centers.

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. Introduction

The session management server is an optional Security Access Manager component that runs as a WebSphere service. It manages user sessions across Security Access Manager servers, ensures that the session state remains consistent across the participating servers, and allows for the implementation of session policy across the participating servers.

The session management server allows Security Access Manager WebSEAL and the Security Access Manager Plug-in for Web Servers to share a unified view of all current sessions, and permits authorized users to monitor and administer user sessions. The session management server also makes available session statistics, provides secure and high-performance failover, and provides single sign-on capabilities for clustered environments.

Designed to manage sessions across clustered Web server environments, the session management server can adapt to complicated deployment architectures.

The session management server is a J2EE application that runs on the WebSphere server, or within a WebSphere cluster.

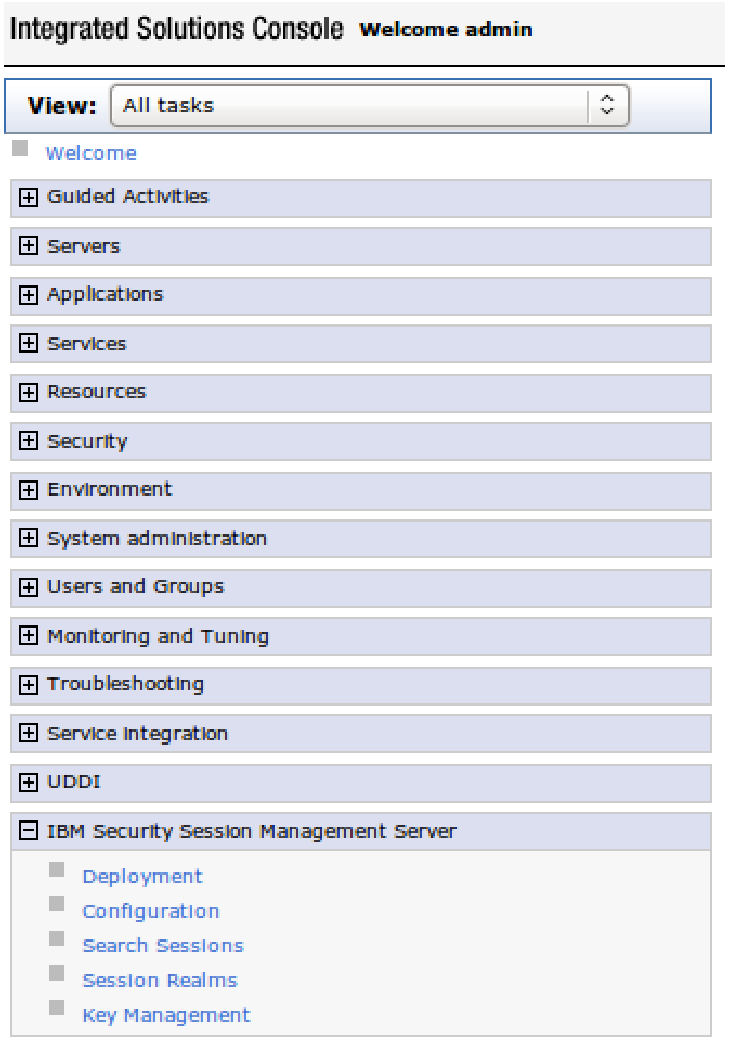
Session management server administration options

You can use any, or all of the following tools to administer sessions:

pdadmin
Details of **pdadmin** commands that can be used for session management are described in "SMS pdsmsadmin and pdadmin commands" on page 32.

pdsmsadmin
The **pdsmsadmin** command line uses the SOAP protocol to communicate directly with a session management server installed on WebSphere Application Server.

The Session Management Server console
The Session Management Server console is a graphical user interface on the WebSphere Application Server, and is installed as an extension to the WebSphere Integrated Solutions Console. Figure 1 on page 2 shows the extended main menu items.



View: All tasks

Figure 1. Integrated Solutions Console session management server extension menu

For more information about the specific administration tasks, see Chapter 3, “Session management server usage,” on page 33.

Additional administrative tasks are done by using the session management server utilities that are described in Appendix B, “SMS utilities,” on page 75.

When you decide which administration options are best suited to your session management server environment, see “Increasing the heap size of host WebSphere Application Servers” on page 9.

Session management server features

Session management server functions generally require initial configuration, and minimal user intervention thereafter.

Session information consistency

The session management server is a centralized Web service that maintains session information across Web security servers. Session data includes user credentials, session timeout information, and other data used by Security Access Manager to track the state of all user sessions.

For example, this consistency applies to users' authentication level so that when a user steps up to a higher authentication level, all replicated WebSEAL and Web server plug-in servers automatically have the updated credential available.

Similarly, when a user session times out, either due to inactivity or session lifetime expiry, the user session ends across all servers.

Cluster-wide login policy enforcement

The centralized view of all user sessions that are maintained by the session management server provides a single point from which to enforce user identity-based session policies.

The maximum number of concurrent sessions a user has across the cluster can be limited by policy. The session management server enforces the policy that can either be:

- Set to refuse a new session that exceeds the maximum, or
- Set to replace a current session with the new session

Failover

Earlier versions of Security Access Manager required the use of a failover cookie to approximate the session replication capabilities of the session management server.

When configured to use the session management server, you do not have to use the failover cookie. The session management server provides a more consistent and complete view of user sessions across sets of replicated Web security servers.

Session information management

The session management server records a variety of session information. Having session information available in a central location offers the ability to manage and monitor sessions across servers. The session management server records the following session information:

- Concurrent login information
- Session statistics information, such as the number of users logged in

Authorized users have access to this information and can use this information to promote system security.

Support for multiple instances

Security Access Manager supports multiple session management server installations on a single WebSphere Application Server. Each installation is called an *instance*. Each session management server instance can contain one or more session realms, and each session realm can contain one or more replica sets. These terms and concepts are explored more fully in “Session management server architecture” on page 4.

Using the session management server administrative tools, you can view all available instances, deploy and configure new instances, or swap from one instance to another to perform administrative tasks.

Limited session realms

In Security Access Manager version 7.0, you can limit the maximum number of sessions for a particular session realm.

After the maximum number of sessions is reached, further session requests are denied until the number of sessions drops below the set threshold.

Session management server architecture

The session management server is built to run on WebSphere Application Server and WebSphere Network Deployment.

The session management server supports communications using the IBM WebSphere Web server plug-in (IBM HTTP Server) or Microsoft IIS (on Windows). During configuration you are prompted to provide the WebSphere port number. This port number should be the port for WebSphere communication using the WebSphere Web server plug-in.

Figure 2 shows the Security Access Manager blade and WebSphere interface.

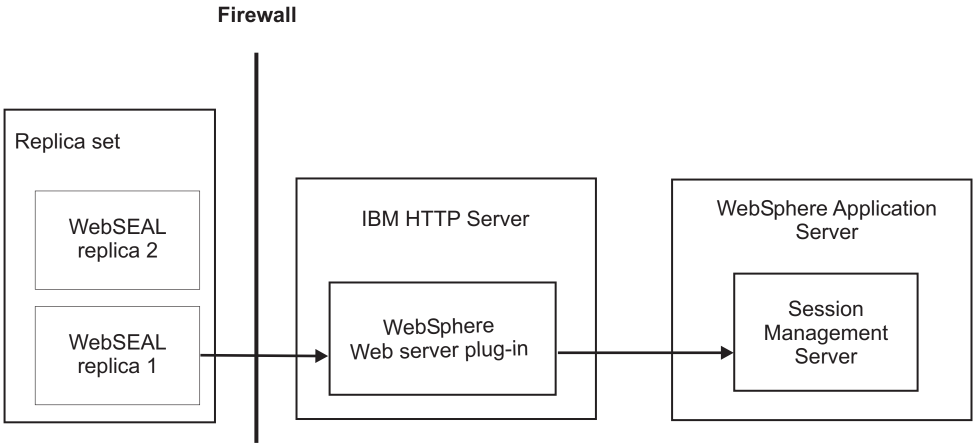

Firewall
Replica set
WebSEAL replica 2
WebSEAL replica 1
IBM HTTP Server
WebSphere Web server plug-in
WebSphere Application Server
Session Management Server

Figure 2. Simple session management server architecture.

The session management server is useful in environments that have replicated Web security servers. Web servers secured by Security Access Manager are replicated to provide high availability and load balancing. Alternatively, single servers in a cluster of servers can form part of a larger application.

When servers are clustered for reasons of high availability and load balancing, the content of the participating servers is identical. Often a load balancer is used to distribute the Internet traffic across each replicated server.

The term *replica set* is used to refer to a collection of replicated WebSEAL (or Web server plug-in) Web security servers. Replicated servers within a replica set serve the same content, are configured the same way, and enforce the same security policies.

Figure 3 and Figure 4 show typical architectures for failover and load balancing with either Security Access Manager Plug-in for Web Servers or WebSEAL. In both cases, WebSEAL and the plug-in are replicated, and both use the session management server to maintain session information across the clustered servers. From a user point of view, a session exists as a single entity across each environment.

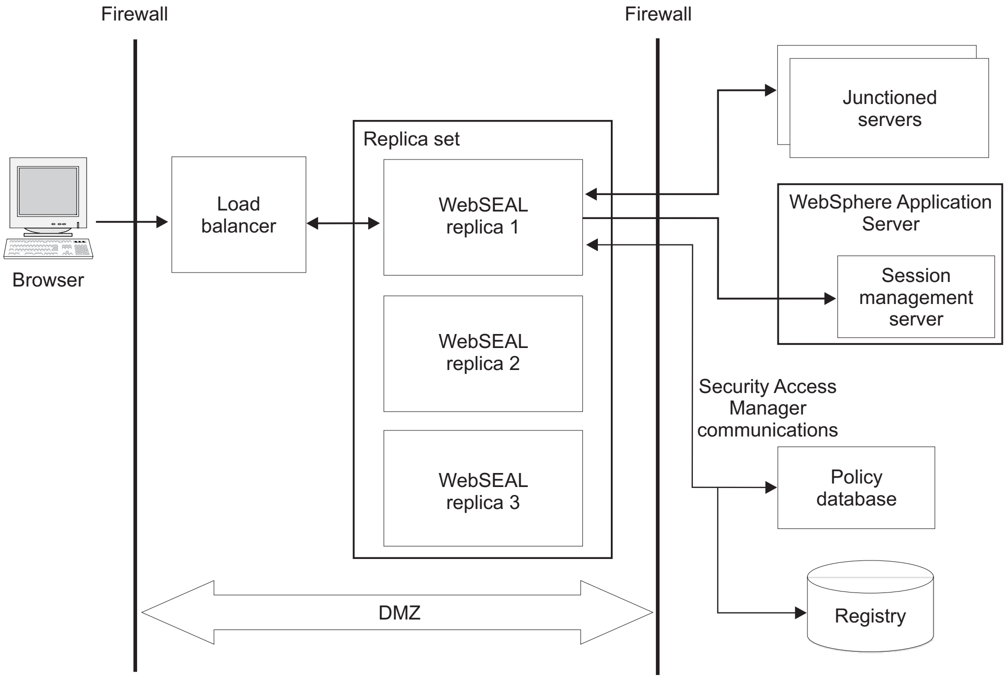


Figure 3. A typical session management server architecture with WebSEAL

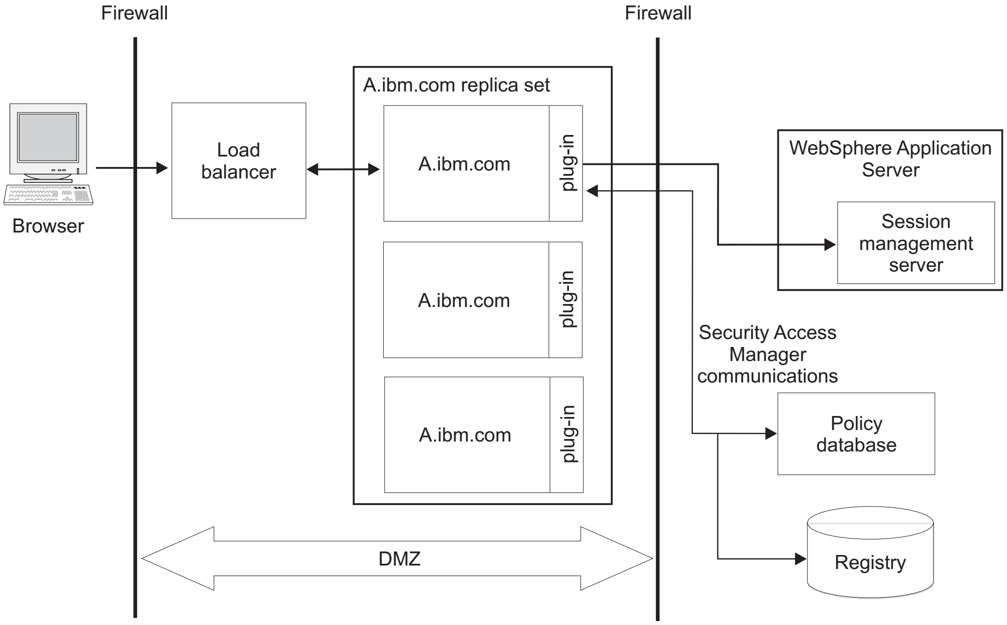


Figure 4. Basic session management server architecture with Security Access Manager Plug-in for Web Servers.

The protected servers shown in Figure 3 and Figure 4 might also be used to host content for different (yet related) Web sites, or each server might form part of a larger, single application. When a user who is accessing these servers perceives them as a single application that requires a single login and consistent concurrent session policy, the session management server can be used to provide secure access across all of the servers.

The extent of a session within a server cluster is referred to as the session realm. The session management server can provide a seamless single sign-on experience across a *session realm*. Servers are added to or removed from session realms by configuration within WebSEAL or the Security Access Manager Plug-in for Web Servers.

Figure 5 shows a representation of a session realm. A session realm consists of one or more replica sets and the user session is replicated across the entire session realm. When users log in, they are considered logged in to the entire session realm. Concurrent session policy is applied across the entire session realm. If a user who is limited to a single concurrent session logs in to one replica set within the realm and then tries to log in to another replica set within realm, the second login is denied.

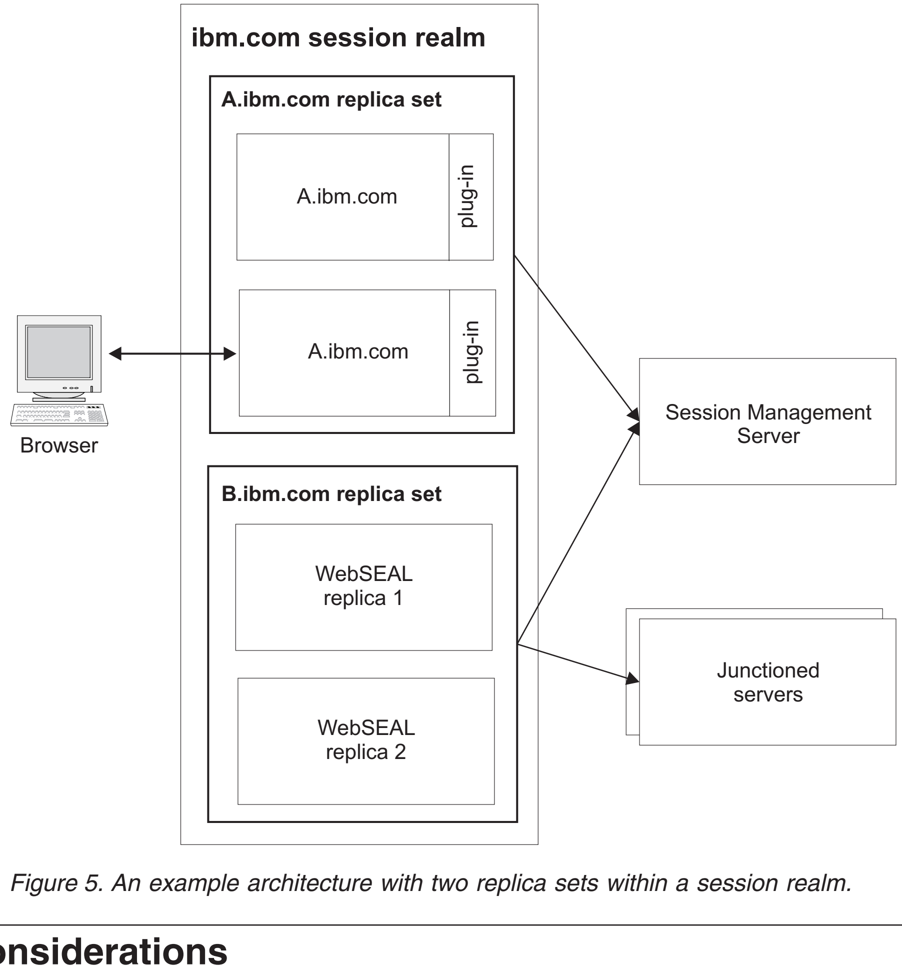


Figure 5. An example architecture with two replica sets within a session realm.

Deployment considerations

Set up the Security Access Manager environment before you install and configure the session management server. You must have a thorough understanding of the structure of your session realms and associated replica sets before you start with the configuration. Another configuration prerequisite is that you must know the replica sets which are not assigned to a specific session realm.

Note: If you are planning to deploy the Session Management Server in a WebSphere version 8.0 environment, the SMS requires WebSphere eXtreme Scale version 8.5.0.1. See the Setting up a session management server section of the *IBM Security Access Manager for Web Installation Guide* for details about the prerequisites.

Before installation and configuration, decide whether you want replicated session management server instances (WebSphere Network Deployment only). Having more than one session management server that serves your Security Access Manager sessions can provide a failover capability and improve performance.

To use the **pdadmin** command for administration purposes, you must install and configure the Session Management Server Command Line Extension component to a Security Access Manager authorization server. By contrast, **pdsmsadmin** does not require an authorization server because it communicates directly to WebSphere Application Server.

The **pdadmin** command line provides server tasks that communicate with the session management server to conduct administrative operations. As server tasks, they are suitable for use in custom administrative applications of your own that can be developed by using Security Access Manager administration APIs. Such API development can be done only with **pdadmin**, not **pdsmsadmin**.

An authorization server is required to use:

- The credential refresh capabilities of the session management server.
- The certificates that are issued by the Security Access Manager policy server

The certificates that are issued by the policy server provide authentication between the session management server and its client applications. For more information, see “Configuration of secure communications” on page 9.

When an authorization server is required, it is typical to deploy an authorization server to each machine that hosts an instance of the session management server.

Figure 6 on page 8 shows a basic structure of the various administration interfaces for the session management server.

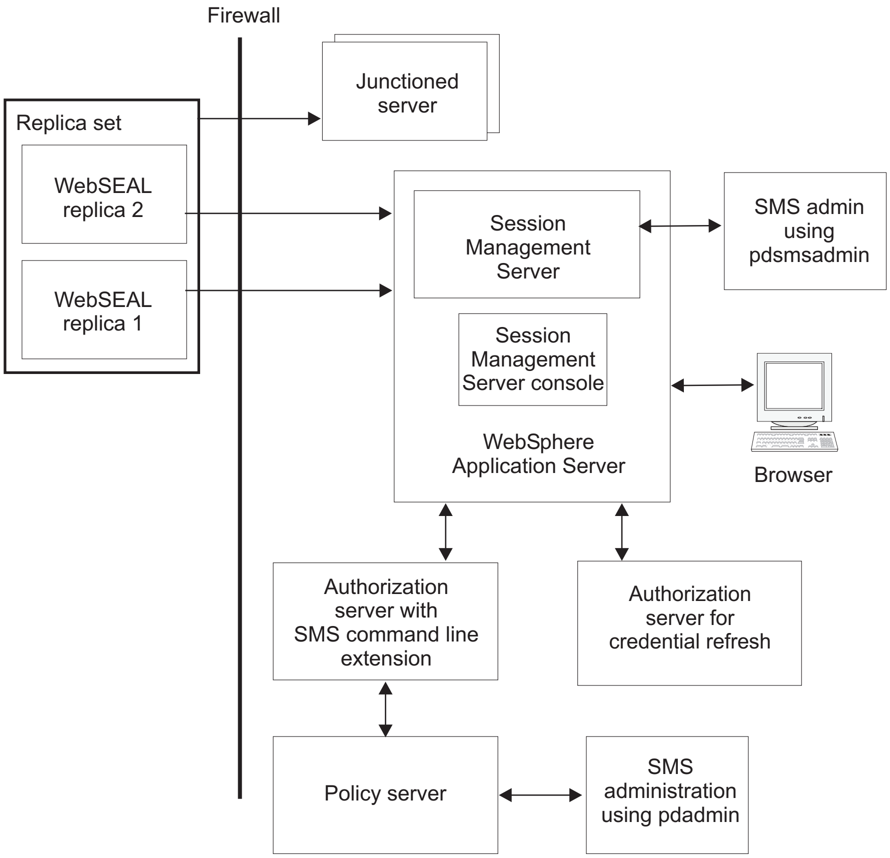

Firewall
Junctioned server
Replica set
WebSEAL replica 2
WebSEAL replica 1
Session Management Server
SMS admin using pdsmsadmin
Session Management Server console
WebSphere Application Server
Browser
Authorization server with SMS command line extension
Authorization server for credential refresh
Policy server
SMS administration using pdadmin

Figure 6. Session management server administration architecture

When deployed, the session management server is a critical Security Access Manager component. It must remain highly available so that the client does not become unavailable. The session management server must therefore be run in a clustered environment which consists of at least two cluster members. If all cluster members become unavailable, the session data that are maintained by the session management server are lost.

For continued service when you restart the SMS cluster, use only the WebSphere Application Server **ripple start** option if there are three or more cluster members. If there are only two cluster members in the cluster, use a manual start and stop of each cluster member one at a time. This process ensures that the second cluster member is brought down only when the first one is back up and running again.

To help ensure high availability, restart the SMS cluster only under conditions of low activity, when there are minimal Security Access Manager sessions open. It is also important to ensure that the catalog service is always running and any container servers can connect to it during a restart.

A WebSphere core group must be set up, and dedicated to the session management server. This core group must include only those cluster members that contain an instance of the session management server. To avoid unnecessary data replication across the network, the WebSphere Application Server Network Deployment Manager must not be a member of the SMS core group.

For more information about WebSphere Application Server Core Groups, see the WebSphere Application Server Information Center.

Increasing the heap size of host WebSphere Application Servers

In some situations, you might need to increase the heap size of the hosting WebSphere Application Servers. This will usually be required if a large number of concurrent sessions will be managed by the Session Management Server.

Procedure

1. Open the Integrated Solutions Console.
2. On the left hand side, expand the **Servers** heading and click **Application servers**.
3. Click on the name of the server you wish to modify.
4. Under the **Server Infrastructure** heading, expand the **Java and Process Management** heading and click **Process Definition**.
5. Under the **Additional properties** heading, select **Java Virtual Machine**.
6. In the **Maximum Heap Size** text box, specify the new maximum heap size.
7. Click **OK**.
8. Click **Save** to save the changes.
9. Restart the application server for the changes to take effect.

Modifying the Deployment Manager heap size

You might need to modify the maximum heap size of the Deployment Manager.

Procedure

1. Open the Integrated Solutions Console.
2. On the left hand side, expand the **System administration** heading and click **Deployment manager**.
3. Under the **Server Infrastructure** heading, expand the **Java and Process Management** heading and click **Process Definition**.
4. Under the **Additional properties** heading, select **Java Virtual Machine**.
5. In the **Maximum Heap Size** text box, specify the new maximum heap size.
6. Click **OK**.
7. Click **Save** to save the changes.
8. Restart the deployment manager server for the changes to take effect.

Security considerations

There are two steps to applying security when using the session management server:

1. Configuring secure communications between the WebSphere server that hosts the session management server and the session management server client applications.
2. Enabling J2EE security on the WebSphere server that hosts the session management server including defining the membership of the session management server application roles.

Configuration of secure communications

You can configure secure communications among the following components. See Figure 6 on page 8 for details:

- The Security Access Manager web security and authorization servers and the WebSphere web server plug-in that accesses the session management server.
- The WebSphere web server plug-in and the WebSphere server itself.

Configuring SSL for these connections can result in a small performance degradation. However, session information is sensitive and it is important to keep it secure. For secure communication, configure SSL for these connections.

Note:

1. When security is enabled in a WebSphere environment, you must use the same user registry that is used by Security Access Manager. You must also add the `sms-administrator` role to the list of users or groups who need access to the Session Management Server console. To complete this task, select **Users and Groups** in the WebSphere ISC.
2. For WebSphere clustered environments, changes to keystore and truststore must be consistent across every server in the cluster. See the WebSphere documentation for assistance.

The following options can be used to achieve the secure communications:

- Leave the connections unsecured.
- Configure SSL between the connections by using the certificates that are issued by the Security Access Manager policy server during the configuration of each Security Access Manager server.
- Configure SSL between the connections by using certificates that you provide yourself.

Configuring secure communications by using policy server-issued certificates

When a Security Access Manager server is configured, the policy server issues a certificate that the server uses to authenticate to the Security Access Manager infrastructure. These same certificates can be used to authenticate SSL communications between these servers and the session management server.

To use these certificates, the web server that hosts the WebSphere Web server plug-in must be configured with the following certificates:

- A certificate that is issued by the Security Access Manager policy server.
- The Security Access Manager policy server CA certificate.

The WebSphere Web server plug-in is used to communicate with the session management server. The certificate that is issued by the policy server ensures that the client applications trust the Web server. The policy server CA certificate ensures that the Web server trusts the client applications.

To obtain a certificate from the policy server for use by the web server, you can run the **svrsslcfg** utility on a machine that is configured with the IBM Security Access Manager runtime. For example, a machine which runs WebSEAL or an authorization server or the policy server. You can run the utility as follows:

```
touch /tmp/was-pi-sms.conf
svrsslcfg -config -n was-pi-sms -h hostname_of_web_server -l no -a no \
-f /tmp/was-pi-sms.conf -d /tmp -r 0 -s remote
```

After you run the utility, it creates `was-pi-sms.kdb` and `was-pi-sms.sth` files in the `/tmp` directory. The `was-pi-sms.kdb` database file contains the certificate to be used by the Web server. The `was-pi-sms.sth` stash file contains the password that is needed to access the certificate file.

If you are using IBM HTTP Server as the Web server, these files can be configured directly to IBM HTTP Server when you enable it for SSL. For Web servers that do not recognize the CMS key file format, use the IBM Java iKeyman tool. The iKeyman tool converts the key file into a format that the Web server can understand.

Note: iKeyman is packaged with IBM Java version 6 or later. For more information, see the *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 and 7: iKeyman User's Guide for version 8.0* at http://download.boulder.ibm.com/ibmdl/pub/software/dw/jdk/security/60/iKeyman.8.User.Guide.pdf.

The subject distinguished name (DN) specified in certificates that are issued by the policy server do not correspond to user entries in the user registry. When you use certificates to authenticate to WebSphere, WebSphere requires the subject DN to map to the user DN in the WebSphere user registry.

To overcome the problem, the session management server provides a Trust Association Interceptor (TAI). The Trust Association Interceptor maps the subject DN of a policy server issued certificate to the DN of the user to whom the certificate corresponds. Part of the session management server configuration process is to enable Trust Association Interceptors.

Note: Security Access Manager is compatible with all versions of the Trust Association Interceptor.

For more information about WebSphere TAIs, see the WebSphere documentation.

Configuration of secure communications using user-provided certificates

You can use certificates other than those issued by the Security Access Manager policy server for either end of the communications between the session management server and its client applications.

You can obtain such certificates either from an external source, your own PKI infrastructure or by creating self-signed certificates using a tool like the IBM Java iKeyman tool.

Note: iKeyman is packaged with IBM Java version 6 or later. For more information, see the *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 and 7: iKeyman User's Guide for version 8.0* at http://download.boulder.ibm.com/ibmdl/pub/software/dw/jdk/security/60/iKeyman.8.User.Guide.pdf.

The only requirement is that each end of the communication has a certification authority (CA) certificate that can verify the validity of the certificate that is presented by the other end of the communication.

Configuring secure communications between the WebSphere Web server plug-in and the WebSphere server

When you install the WebSphere Web server plug-in, you can use a key file that contains sample certificates to communicate between the plug-in and the application server.

Note: Do not use the sample certificates in an environment that requires secure communications between the plug-in and the application server.

The WebSphere documentation describes how to create your own certificates for the plug-in to communicate with WebSphere. You can also configure the plug-in to use with the same certificates that are used by the Web server to communicate with the session management server client applications.

When you are using certificates that are issued by the Security Access Manager policy server, the session management server configuration process creates a WebSphere SSL repertoire. It also consists of policy server-issued certificates. For details on changing the SSL repertoire that is used in communication with an application, see the WebSphere documentation.

Configuration of session management server authorization

The session management server uses the J2EE role-based authorization model that is provided by WebSphere to authorize operations that are requested by its client applications. WebSphere J2EE security must be enabled before the session management server operations are authorized.

The WebSphere documentation describes how to enable J2EE security. The requirements of the session management server are:

- Lightweight Third Party Authentication (LTPA) is enabled as the authentication mechanism.
- The subject DNs of certificates that are used to authenticate to WebSphere by session management server client applications, correspond to the DN of users in the WebSphere registry. It requires use of an LDAP-based user registry. For example, IBM Tivoli Directory Server or Microsoft Active Directory.
- To avoid replication of user data between Security Access Manager and WebSphere, you must configure WebSphere to use the same user registry as Security Access Manager.

After security is enabled, access to the roles used to authorize access to the various session management server operations must be granted.

The session management server defines the following interfaces:

Session management interface
The interface that is used by Web security servers to create, retrieve, modify, and end user sessions.

Session administration interface
The interface that is used by administrative applications to administer the session management server and the sessions it maintains.

Access to these interfaces is authorized separately.

Authorizing users and groups to access the session management interface

You can grant access for users to the session management interface either directly or by group membership.

About this task

Access to the session management interface is controlled by the **sms-client** role. Session management server client applications authenticate to WebSphere using their certificate. This certificate corresponds to a WebSphere user.

Define groups of users and assign these groups to roles, rather than assigning users to roles directly. This approach makes changes in role assignments simpler to manage because you just need to change the group membership.

Procedure

1. Login to the WebSphere administration console.
2. Select **Applications** > **Enterprise Applications**, then the instance name.
3. Click **Security role to user/group mapping**.
4. Select the **sms-client** role check box, and click either **Lookup groups** or **Lookup users**.
5. Look up the groups and users you want to assign the **sms-client** role and add the required groups or users.
6. Click **OK**.
7. Save the WebSphere configuration changes. This automatically restarts the session management server application.

Session administration interface authorization

The authorization of accesses to the session administration interface is slightly different. Because administration operations are requested by either **pdsmsadmin** or the Security Access Manager authorization server, it is the user identity for these processes that authenticates to WebSphere.

Users logging in to **pdsmsadmin** or the Security Access Manager authorization server pass user identity information on to the session management server, indicating the identity of the real user who is requesting the administration operation (**sec_master**, for example). As such, the identity from the client certificate for either **pdsmsadmin** or the Security Access Manager authorization server acts as a delegate of the real user requesting the operation.

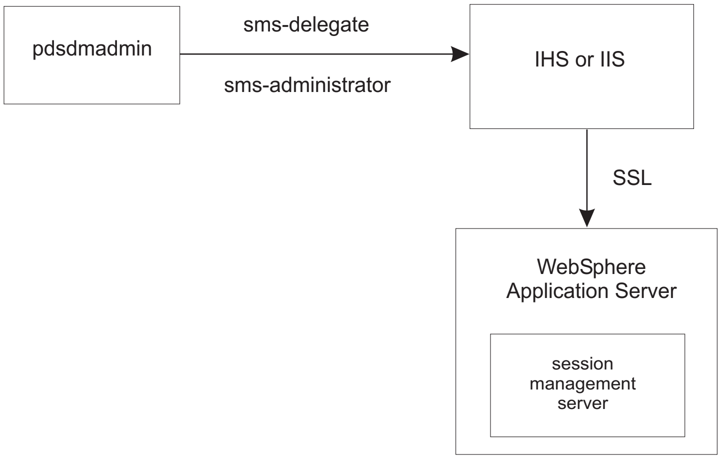


Figure 7. Security levels for pdsmsadmin communications

The preceding diagram shows two levels of security between **pdsmsadmin** and the IHS or IIS server:

- Communication channel security is provided by the client certificate. To ensure that only trusted entities can specify the identity of the real user who is requesting the administration operation, the first authorization that occurs checks that the user authenticated to WebSphere has the **sms-delegate** role, indicating that the user can be trusted to reliably specify the real identity of the user requesting the administration operation.
- The User ID provides application security. If the first security test is passed, then the real user identity performs a second authorization. The real user must possess the **Administrator** and **sms-administrator** roles to perform session management server administrative tasks.

For example, consider a deployment where:

- An authorization server is running on host `server1.ibm.com`, with a user name of `ivacld/server1.ibm.com` (configured with the session management server command line extension)
- A user called `sms-admin` logs in to **`pdadmin`** to perform administration operations against the session management server

For the administration operations to succeed, the `ivacld/server1.ibm.com` user must have the **sms-delegate** role and the **sms-admin** user must have the **Administrator** and **sms-administrator** roles. Access to these roles is granted in the same manner as access to the **sms-client** role previously described.

WebSphere usage of LDAP-based user registries

WebSphere allows configuration of a base distinguished name (DN). This is used as a starting point for all searches of the registry.

Configuration of a base DN allows you to login, for example to the WebSphere administration console, without specifying the full DN of the user you are logging in as.

However, when a base DN is specified, all DNs used for authentication must be a child of this base DN. If not, authentication will fail.

For example, if you have configured a base DN of `C=US, O=IBM`, then DN of all users must begin with `C=US, O=IBM`. Users with a DN of `C=AU, O=IBM` will not be able to authenticate.

Some Security Access Manager administrative users have Security Access Manager specific DNs. In particular, the administrative user **sec_master** will generally not have a DN of the form used by other users in the registry. To perform session management server administrative operations as the user **sec_master** you must therefore ensure no base DN is configured for the WebSphere LDAP user registry.

Similarly, if you are using Security Access Manager policy server issued certificates for authenticating SSL communications between the session management server and its client application, you must ensure no base DN is configured for the WebSphere LDAP user registry. Security Access Manager certificates correspond to registry users with a DN in a Security Access Manager specific part of the user registry.

Configuring the security role membership for SMS administrator

When WebSphere Administrative Security is enabled, you must configure the SMS administrator to be a member of the following roles:

- **Administrator**
- **sms-administrator**

You can configure security role membership for an SMS administrator for the WebSphere Application Server, versions 7.0 and 8.0 by using the WebSphere Integrated Solutions Console.

Configuring the SMS administrator role for WebSphere Application Server, versions 7.0 and 8.0

In WebSphere Application Server, version 7.0 and version 8.0, you cannot assign new roles to the primary administrative user. To complete SMS administration tasks, create a separate administrative user to administer the SMS.

About this task

You might have to configure the SMS administrator role for WebSphere Application Server, version 7.0 or version 8.0.

The set of security roles for SMS administrators, clients, and delegators are pre-defined. The roles are only accessible after you enable administrative security. The security roles are **sms-administrator**, **sms-delegrator** and **sms-client**.

You use predefined security roles to authorize access to the various SMS operations. To complete SMS administrative tasks, you must configure the SMS administrator to be a member of **Administrator** and **sms-administrator** roles.

Procedure

1. Create an administrative user and group. You must create a user in LDAP that you can assign the roles for administering SMS and WebSphere Application Server.
   a. Start the **`pdadmin`** command-line utility.

b. Enter the command to create the administrative user and group. For example:

   The following example creates a user that is called `wasadmin` and adds the user to a new group that is used to assign **sms-administrator** access.

   ```
   pdadmin sec_master> user create wasadmin "cn=wasadmin,o=example,c=us"
    wasadmin wasadmin password
   pdadmin sec_master> user modify wasadmin account-valid yes

   pdadmin sec_master> group create sms-admin-group
    "cn=sms-admin-group,o=example,c=us" sms-admin-group
   pdadmin sec_master> group modify sms-admin-group add wasadmin
   ```

2. Enable administrative security in WebSphere Application Server. Enabling administrative security in WebSphere Application Server, secures the WebSphere Application Server. You can also configure the necessary security roles for SMS.
   a. Click **Security** > **Global Security**.
   b. Select the **Enable administrative security** check box.
   c. Configure the user account repository. By default, the **Current realm definition** is set to **Local operating system**.
      1) Under **Available realm definitions**, select **Standalone LDAP registry**.
      2) With **Standalone LDAP registry** selected, click **Configure**.
      3) Configure the LDAP registry.

         Note: Avoid the use of **Base distinguished name** because it limits the users and groups that SMS can use for roles.
   d. Click **OK**.
   e. Ensure that the **Current realm definition** is set to **Standalone LDAP registry**. In the **Global Security** page, for **Available realm definitions** with **Standalone LDAP registry** selected, click **Set as current**.
   f. Click **Apply**.
   g. Ensure that the **Enable administrative security checkbox** is selected.
   h. Click **OK**.
   i. Save changes to the master configuration.
   j. Restart WebSphere Application Server.
   k. Log on to the WebSphere Application Server administrative console with the WebSphere administrative user. For example: `wasadmin`
3. Configure security role membership for the SMS administrator. Setting security roles ensure that only appropriate clients are allowed to be SMS administrators, clients, and delegators.
   a. In the WebSphere Application Server administrative console, click **User and Groups** > **Administrative group roles**.
   b. Click **Add**.
   c. Under **Role(s)**, select both **Administrator** and **sms-administrator**.
   d. Ensure that **Map Groups As Specified Below** is selected.
   e. In **Search String**, enter `*`.
   f. Click **Search**. A list of all the available groups are displayed.
   g. Select the SMS administration group to map the security roles to. For example: **cn=sms-admin-group,secAuthority=default@example:389**.
   h. Add the SMS administration group to the **Mapped to role** list.

Note: If the role mapping is not completed successfully, the WebSphere administrative user, `wasadmin` cannot see the **Tivoli Session Management Server** portlet in the WebSphere Application Server administrative console.

i. Save the settings and log out.

Results

You can log on to the WebSphere Application Server administrative console with the WebSphere administrative user account, `wasadmin`, to verify that the **Tivoli Session Management Server** portlet is visible.

WebSEAL and Plug-in for Web Servers configuration

The participating Security Access Manager blades (WebSEAL, the Plug-in for Web Servers, or both) need to be configured to use the session management server for managing sessions. Configuration of these products is not covered in this document and is instead detailed in the respective guides for these products:

- *IBM Security Access Manager: WebSEAL Administration Guide*
- *IBM Security Access Manager: Plug-in for Web Servers Administration Guide*

Configuration of these components is the last step in session management server configuration. However, it is important to realize that configuration of these entities is required. Therefore, they need to be installed and running before the session management server can operate. For complete installation instructions for Security Access Manager components, see the *IBM Security Access Manager for Web: Installation Guide*.

Single sign-on with the session management server

The session management server provides a single sign-on (SSO) capability across replica sets in a session realm. This SSO is based on a domain cookie set by the Security Access Manager blade. Use of a domain cookie requires that all of the replica sets be peer DNS domains, so that when set by a member of one replica set in the realm the browser will submit the cookie to the other replica sets in the realm.

If SSO across DNS domains is required, an e-Community single sign-on (eCSSO), cross-domain single sign-on (CDSSO), or External Authentication Interface (EAI) solution should be considered. Details on these cross-domain SSO approaches are documented in the *IBM Security Access Manager: WebSEAL Administration Guide* and the *IBM Security Access Manager: Plug-in for Web Servers Administration Guide*.

Note: Single sign-on performed using anything other than the session management server domain cookie results in multiple sessions for the user being created at the session management server. To apply concurrent session policy, you need to take this into account when designing your replica sets and session realms.

If each replica set is configured as part of the same session realm, a single user, signed on to multiple replica sets using eCSSO, will have multiple sessions. If single sign-on can occur between replica sets using a method other than the session management server domain cookie and you want to make use of concurrent session policy in the replica sets, those replica sets should not be part of the same session realm.

That is, the following conditions must apply before it is necessary to split the replica sets into different realms:

- Requires concurrent session policy
- Requires the ability to single sign-on between replica sets using a method other than session management server domain cookies

The session management server single sign-on facility across replica sets within a DNS domain removes the need for the failover cookie.

A single session ID is used across each replica set to represent the user's single session across the entire session realm. A session realm can consist of replica sets of any kind (for example, WebSEAL and Plug-in for Web Servers).

If access privileges permit, the client (as illustrated in Figure 5 on page 6) can move between any of the servers in the session realm (`ibm.com`, in the figure) without the need to re-authenticate.

The session management server allows for single sign-off across replica sets. Signing out of one replica set ends the session across the entire realm.

Back-end storage mechanisms

You can configure the Session Management Server to use one of three different back-end storage mechanisms.

The deployment configuration options on the target computer or cluster of the SMS deployment determine which storage mechanism is used.

Single server

Two back-end storage mechanisms are available for a stand-alone SMS server:

- In-memory
- Database

In-memory

The in-memory storage mechanism is the default mechanism for a single stand-alone SMS server.

In-memory storage is not suitable for a production environment because this mechanism does not scale and is not fault tolerant. In-memory storage is a good option to use when you demonstrate or review the capabilities of the SMS in a proof-of-concept environment.

Database

You can store the SMS session data in a database. To enable this storage mechanism you need to specify `yes` to the `Enable database storage` option when deploying the SMS using the **smscfg** tool. Only a single-server SMS deployment supports storing session data in a database, which makes this storage mechanism less viable for a highly available production environment.

Note: Database storage can be used for last login information, even in a clustered environment.

Clustered server

For a clustered SMS environment, WebSphere eXtreme Scale storage is the only available back-end storage mechanism.

WebSphere eXtreme Scale

In a clustered environment, the SMS session data is stored using WebSphere eXtreme Scale, which is an IBM product included with Security Access Manager. WebSphere eXtreme Scale is a scalable data grid that can replicate data across JVM instances to ensure high availability. This configuration is the most appropriate for a production environment because of its high availability and scalability.

Chapter 2. Configuration

This chapter explains how to deploy, configure, and unconfigure session management server instances.

It contains the following sections:

- "Session management server installation"
- "Deploying the session management server" on page 22
- "Session management server configuration" on page 24
- "Configuration of command line extensions" on page 28
- "Unconfiguration of the session management server" on page 29
- "Deployment and configuration of more instances" on page 29
- "Adding a member to an SMS cluster" on page 32

Session management server installation

For complete session management server installation details, including pre-installation considerations and requirements, see the "Session Management Server" section of the *IBM Security Access Manager for Web: Installation Guide*.

Installation of the session management server involves:

- Setting up a session management server,
- Setting up the session management command line(s).

Both tasks can be performed using the command line, script files, or Launchpad (Windows only). Following installation, the session management server application must be deployed and configured. For deployment details, see "Deploying the session management server" on page 22. For configuration details, see "Session management server configuration" on page 24.

Once you have installed, deployed and configured the session management server, administrative tasks for session management can be performed using any and all of the following administrative tools:

- The **pdadmin** command line extension
- The **pdsmsadmin** command line extension
- The Session Management Server console, which is installed as an extension to the WebSphere ISC

The availability of these tools will depend on your installed session management components. To administer the session management server with the **pdadmin** or **pdsmsadmin** commands, you must install the PDSMSCLI package. For **pdadmin**, the PDSMSCLI package must be installed on the same system as your Security Access Manager authorization server.

To administer the session management server from the Session Management Server console, you must install the PDSMS package on your WebSphere system. Once you have deployed the Session Management Server console extension, you can use it to deploy and configure additional session management server instances (see "Deployment and configuration of more instances" on page 29).

For full installation details, see the "Installing session management system components" section of the *IBM Security Access Manager for Web: Installation Guide*.

Installing fix pack upgrades

You can use the **smscfg** utility to install Session Management Server fix packs. You can also use this utility to remove fix packs. A complete version history is kept, so you can revert all the way back to the version that was originally installed.

Procedure

1. Install the updated package (installation program, RPM, `pkgadd`, and the rest of files) on your deployment manager (for network deployment) or application server machine (for single servers).
2. For each deployed SMS instance, run the command: `smscfg -action upgrade -instance` *`instance_name`*

Results

This process applies the installed fix pack level to the specified SMS application. During this process, the session management server application is restarted. For clusters spread across multiple nodes, no data is lost.

To remove a fix pack, run the following command:

```
smscfg -action revert -instance instance_name
```

This command reverts to the most recent fix pack that is applied to the specified instance. That is, the version that was installed before the fix pack was applied. Again, the application is restarted.

You can apply a fix pack only if it is more recent than your current application version. To apply an older fix pack, you must first revert to a version older than the fix pack. For example, you might upgrade your SMS instance directly from fix pack 2 to fix pack 4, but later decide that you want fix pack level 3. In this case, you must revert to fix pack 2 before you can upgrade to fix pack 3. However, you can upgrade directly from fix pack 4 to a newer fix pack, such as fix pack 5.

Note: Session Management Server requires WebSphere eXtreme Scale version 8.5.0.1 or later to deploy to a WebSphere Application Server version 8.0 cluster. Use the latest WebSphere eXtreme Scale version if there are stability issues.

Deploying the session management server

The installation process includes deployment of the session management server application. You can use the **smscfg** utility or the Session Management Server console for this deployment.

Notes:

1. Before it can be used, the Session Management Server console extension must be deployed. Run the command:
   ```
   smscfg -action deploy -instance ISC
   ```
2. If you plan to deploy the Session Management Server in a WebSphere version 8.0 environment, the SMS requires WebSphere Application Server version 8.0 FP5 (or later).

3. The use of DB2 as the session storage mechanism in a WebSphere Application Server clustered environment is not supported. However, you can use it to store the last login information.
4. If you intend to use a DB2 database to store login history information, you must create the database before you deploy the session management server application. For details, see the *IBM Security Access Manager for Web: Installation Guide*.
5. A supported version of WebSphere eXtreme Scale must be installed on each node of the WebSphere cluster before the deployment of the SMS. For details, see the Session Management Server section of the *IBM Security Access Manager for Web: Installation Guide*.

Deploying the session management server using the smscfg utility

You can use the **smscfg** utility to deploy the session management server application.

Procedure

1. Prior to running **smscfg**, run the WebSphere `setupCmdLine.bat` or `setupCmdLine.sh` script (depending on your operating system).
2. Deploy the session management server application using the command: `smscfg -action deploy -instance` *`instance_name`*

Results

For more information, see "smscfg" on page 79.

Deploying the session management server using the console

You can use the Session Management Server console to deploy an instance of the Session Management Server application.

Procedure

1. Log in to the Session Management Server console as the Session Management Server administrator.
2. Select **IBM Security Session Management Server**
3. Select **Deployment**.
4. In the **Application name** field, enter the name of the Session Management Server application. This field is required.
5. In the **Target** field, enter the WebSphere Application Server cell element to which the Session Management Server instance will be deployed.
6. In the **Virtual host** field, enter the Web server virtual hosts that will service the Session Management Server application instance.
7. In the **Data source** field, enter the data source to use with the Session Management Server application instance.
8. When you are ready to deploy, click **Deploy**.

Results

Deployment of the application might take several minutes without generating any messages. Click the **Refresh** button to update the current progress. Upon successful deployment, the new instance should be visible.

Session management server configuration

The session management server itself requires configuration to initialize it in the environment.

After you install and configure the session management server and extension components, configure the participating servers in the session realm. The participating servers are either WebSEAL servers, or the Plug-in for Web Servers, or both. Configuration at WebSEAL and the plug-in points these blades at the session management server instance.

You can run the session management server configuration utility, **smscfg** in any of three ways, including a combination of the three:

- Interactively using the **–interactive** `yes` parameter. This approach prompts you to input the required parameters as the utility proceeds.
- Non-interactively using the **–interactive** `no` parameter. This approach requires all parameters to be supplied with the command entry.
- Using the **–rspfile** *path_to_file* response file to store the parameters and the utility to read from the file. You can also record a response file and reuse the stored information for later configuration purposes. Parameters that are entered on the command line take precedence over parameters in the response file.

The discussion of the session management server configuration commands in the chapter assumes that you used the **–interactive** `yes` parameter so that the command prompts you for input. A complete listing and explanation of all session management server utilities is included in Appendix B, “SMS utilities,” on page 75.

Configuration details to gather

You must gather information about the deployment environment before you configure the session management server.

Before starting the configuration, run the **`setupCmdLine`** command to set up the correct execution environment for the tool. In network deployment environments, this utility is in the WebSphere deployment manager `/bin` directory.

Table 1 lists the information that you must gather before beginning the configuration of the session management server.

Table 1. Session management server configuration considerations

Item	Description
WebSphere cluster name	You must decide whether you are deploying the session management server to a WebSphere cluster or a stand-alone server. If deploying to a cluster, you require the cluster name.

Table 1. Session management server configuration considerations (continued)

Item	Description
The WebSphere server host name	Specifies the name of the host where the session management server is deployed. In WebSphere network deployment architectures, the host value is the same as the host for the deployment manager. In WebSphere single server environments, the host value is for the WebSphere server where the session management server was installed. During configuration, default values are offered. These default values are obtained from the `wsadmin.properties` file. **Note:** The installation wizard uses **`wsadmin`** to connect to the WebSphere server and obtain a list of servers. Some additional setup is required for **`wsadmin`** to connect to the WebSphere server when security is enabled using private certificates. The `soap.client.props` file in the *`WAS_path`*`/profiles/default/properties/` directory must be edited to reference the new client key files.
WebSphere user name and password	To enable a secure connection between the participating web servers and the WebSphere server that is hosting the session management server, supply the user name and password required for access to the WebSphere server.
The full path to the truststore and the truststore password	Values for the full path to the WebSphere truststore and the truststore password are only necessary when WebSphere security is enabled. A default value for the truststore location is offered during configuration.
The full path to the keystore and the keystore password	The full path to the keystore and the associated password are only required when WebSphere security is enabled. Default values are offered.
Session realm and replica set structure	The configuration requires you to enter a session realm structure with associated replica sets. It is not necessary to specify any session realms, but you must specify at least one replica set.
Session limit policy	Controls whether the session limit and displacement policy is enabled. The default value is to enable this policy.
Auditing configuration file	Two auditing log files are available, depending on whether you are logging to a file or a CARS service: • *`install_root`*`/etc/textfile_emitter.properties.template` for logging to a file, and • *`install_root`*`/etc/webservice_emitter.properties.template` for logging to a CARS service. These templates must be edited to add configuration details before you configure your SMS server. See the templates for further details.

Table 1. Session management server configuration considerations (continued)

Item	Description
Maximum session lifetime	The maximum session lifetime is the number of seconds that the session management server stores a session. When the maximum session lifetime expires, the session management server removes the session. The server periodically checks for expired sessions every *n*/4 seconds, where *n* is the configured maximum session lifetime. Set this maximum session lifetime to be larger than the SMS client session lifetime that is defined by the `timeout` value in the `session` stanza of the configuration files. This larger setting ensures that any sessions not removed by the clients are removed periodically by the session management server itself. You can disable this option by specifying a value of `0`.
Key lifetime	The lifetime of the key used to sign session management server session IDs requires configuration. After the configured time has elapsed the session management server automatically regenerates the key. This process occurs without the need for user intervention. A reasonable setting for this option must consider security issues associated with key lifetime.
Tivoli® Common Directory (TCD) Logging	You have the option of configuring the Tivoli Common Directory on each host where the session management server is installed. This configuration requires you to enter the full path location of the Tivoli Common Directory. There might be an existing value that you can use. You can find information about the Tivoli Common Directory in the *IBM Security Access Manager for Web: Installation Guide*.
Security Access Manager configuration information	You can enable integration with Security Access Manager. Enable the Security Access Manager integration if you want to use credential refresh functionality. It is not required for other SMS functionality. If you enable the Security Access Manager integration, the configuration process prompts you to enter data that covers the use of Security Access Manager certificates for authenticating clients. Requested details include: • The policy server host name. • The policy server port. The default port is `7135`. • The Security Access Manager administrator ID. The default value is `sec_master`. • The Security Access Manager administrator password. • Authorization server details.

Table 1. Session management server configuration considerations (continued)

Item	Description
Last login parameters	The session management server can be configured to record last login information. This information includes the date and time of the last login (from the current browser) and the number of failed login attempts since the last successful login before the current login. This information is then available for display in a browser if required. A number of parameters are required for configuring last login. The configuration process prompts you to enter: • The name of the database table used to store the last login information. This information is only required if you have selected a data storage type of DB (database). The default database table name is AMSMSUSERINFOTABLE. • The maximum number of entries to be stored in the memory cache for the last login information. The default number of maximum entries is 5000. • The name of the last login JSP file. The default value is lastLogin.jsp. This file is in the installation directory for the session management server.
Data storage type	This parameter defines the registered JDBC database that is used for storing last login and session data, which facilitates recovery if your system fails. If you have chosen to deploy the session management server in a clustered WebSphere architecture, then session information is stored and distributed using WebSphere eXtreme Scale. In a clustered deployment it is not possible to store the session information to a DB2 database. In single server architectures, session information can be stored in a database. The session management server supports the use of DB2 for storage of session information in single server WebSphere architectures. The use of DB2 as the session storage mechanism in a WebSphere Application Server clustered environment is unsupported. You can however use it for storing the last login information. If you choose not to store session information or are using a database other than DB2 (in single server architectures), the session management server cannot recover session information after a failure. Last login data can either be stored to memory or to a database. If you are storing to a database, the selected source must be the same as used to store the session table. The session table can also be stored to a database, memory, or direct to a cluster. The session table can be stored to memory only if deploying to a stand-alone server rather than a cluster.
Client idle timeout	The client idle timeout requires configuration. This timeout is a time value, in seconds, after which the session management server stops communication with a server. For a discussion of this timeout, see "Session information consistency" on page 3.

Configuration utility

Use the **smscfg–actionconfig** utility to configure the session management server. If details are incomplete, the utility displays an interface that prompts you for more information.

The session management server configuration utility is installed in the /bin subdirectory of the session management server installation by default:

Linux and UNIX operating systems
`/opt/pdsms/bin`

Windows operating systems
`C:\Program Files\Tivoli\pdsms\bin`

A log of the configuration progress is stored in the /var/pdsms/log/msg_pdsms_config.log file.

For complete details about this utility, see "smscfg" on page 79.

Configuration of command line extensions

The session management server requires configuration before either the **pdsmsadmin** or **pdadmin** command lines can be used for administrative purposes.

To configure the session management server command line extension, use the **pdsmsclicfg -action config** utility. If integration with Security Access Manager is enabled, you can also use the **pdconfig** utility. Run the command from the server that is hosting the session management server. The command is located in the /bin directory on the session management server installation. For complete information about this utility, see "pdsmsclicfg" on page 75.

There are three ways to configure the session management server command line extension on Windows:

- Security Access Manager Configuration GUI,
- SMS CLI GUI (this is C:\Program Files\Tivoli\PDSMS\bin\pdsmsclicfg.exe), or
- SMS CLI command line (C:\Program Files\Tivoli\PDSMS\bin\pdsmsclicfg-cl.exe).

Which command line to use?

Integration with Security Access Manager is required for **pdadmin**, but not for **pdsmsadmin**. From the perspective of a programmer, **pdadmin** provides access to Security Access Manager administrative APIs, which can be used to execute session management server commands, whereas **pdsmsadmin** does not. **pdadmin** also provides backwards compatibility with version 6.0 of the session management server, for which integration with Security Access Manager was compulsory.

Other considerations

Consider the following points prior to configuration:

- Further configuration requires the name of the server that hosts the session management server and the port number used for communications. If you choose to integrate with Security Access Manager, you also require the name of the authorization server, which hosts the command line extension utility.

You can install more than one session management server for failover and performance reasons. In such cases, make sure that you record the host name, instance and communication port number for each server.

- If integration with Security Access Manager is enabled, the configuration command writes properties to the host authorization server's configuration file, `ivacld.conf`. If this file is not in the default location then the exact location needs to be entered at the time of configuration.
- You need to decide whether to enable SSL communications between the authorization server and the WebSphere server that is hosting the session management server. While SSL can provide additional security for your network, this entails a performance cost that must be considered.

  SSL for this connection can use the authorization server certificates, but this relies on the session management server having also been configured to use the Security Access Manager certificates. Alternatively, you can configure the SSL connection using custom or private certificates, as described in "Configuration of secure communications" on page 9.

  The configuration command requires the following information:

  - The full path to the SSL key file that is to be used to encrypt communications.
  - The full path to the SSL key file stash file.
  - The label of the client certificate in the SSL key file.

Unconfiguration of the session management server

Unconfiguration will remove the session management server from the Security Access Manager Policy Server. This may be useful if you enabled Security Access Manager integration when configuring your session management server instance.

To unconfigure session management server components, use the following utilities:

- **`smscfg` –action unconfig**
- **`pdsmsclicfg` –action unconfig**

For complete details about using these utilities, see "smscfg" on page 79 and "pdsmsclicfg" on page 75 respectively.

Deployment and configuration of more instances

You can use **smscfg** or the Session Management Server console to deploy and configure more instances on WebSphere Application Server.

Deploying a session management server instance using smscfg

You can deploy a session management server instance using **`smscfg`**.

Procedure

- Use the **`smscfg`** command.
- Run this command:

  ```
  smscfg -action deploy [-instance instance_name]
  ```

Deploying a session management server instance using the console

You can deploy a session management server instance using the Session Management Server console.

Procedure

1. Select **deployment** from the main menu of the Session Management Server console.
2. Enter an **Application name** for the new instance in the text field.
3. Use the dropdown menus to select appropriate values for the following:
   - **Target**
   - **Virtual host**
   - **Data source**
4. Click **Deploy**.

Results

Deployment of the application may take several minutes without generating any messages. Click the **Refresh** button to update the current progress. Upon successful deployment, the new instance should be visible.

Configuring a new instance of the session management server extension

After a successful deployment, you must configure the new instance of the session management server extension.

Procedure

1. Select **configuration** from the main menu of the Session Management Server console.
2. Select the check box next to the instance that you want to configure. If the new instance is not visible, click **Update SMS instance list**.
3. Click **configure** to proceed. The configuration process involves providing the following information:

   Session realms and Replica sets

   In the **Configure session limit policy** section, select the **Enable session limit policy** check box if you want to enforce the session limit and displacement policy.

   In the **Configure session realms** section, enter a value in the **Session realm name** field. Specify a new session realm name to create a session realm, or modify an existing session realm by specifying a session realm name that exists. To limit the maximum sessions for the session realm:

   a. Select the **Limit maximum sessions for this session realm** check box.
   b. Enter a **Maximum sessions** limit value.

   Click **Update session realms**.

   In the **Configure replica sets** section, set the **Session realm name** and enter a value in the **Replica set name** field. To create a replica set, specify a replica set name that does not exist and the appropriate session realm, and then click **Update replica sets**. To modify an existing replica set, specify the name of the existing replica set and the modified session realm name and then click **Update replica sets**.

   Note: Clients with the same configuration can connect to the same replica set, but clients with different configurations must connect to different replica sets.

Click **Next** to proceed.

Database storage

Select the **Enable database storage** check box if you want to store session information in a database.

Note: This option is available only if you are configuring an SMS instance in a stand-alone server deployment.

IBM Security Access Manager integration

Select the **Enable IBM Security Access Manager integration** check box to enable the integration.

Click **Next** to proceed.

If you are using the IBM Security Access Manager integration, you must provide additional configuration details:

a. IBM Security Access Manager Policy Server configuration. Enter the **Policy server host name**, **Policy server port**, and **Domain**.

b. Click **Next**.

c. Credential refresh information. Configure the Security Access Manager refresh rules. Each rule comprises an **Attribute refresh operation** (`preserve` or `refresh`) and an **Attribute pattern** that specifies the credential attribute. For example, `tagvalue_username`. After you define these details for each rule, click **Add rule** to add the rule to the list.

   When a credential attribute is updated during a session, Security Access Manager processes these rules in order and applies the first matching rule. You can click **Move up** and **Move down** to change the order of the rules.

d. When the refresh rules are complete, click **Next**.

e. IBM Security Access Manager Authorization Server configuration. Enter the **Authorization server host name**, **Authorization server port**, and **Rank** for each server and click **Update authorization servers** to add it to the list.

f. After you have completed the authorization server configuration for your environment, click **Next**.

Last login recording

Select the check box to enable the recording of last login information. Click **Next** to proceed.

Tivoli Common Directory (TCD) logging

Select the check box to enable Tivoli Common Directory logging. Accept the default path in the **Log directory** field, or specify a new path. Click **Next** to proceed.

Auditing

Select the check box to enable auditing. Click **Next** to proceed.

Timeouts

Accept the default values for **Client idle timeout** (600 seconds), **Key lifetime** (186 days), and **Maximum session lifetime** (7200 seconds), or specify new values. Click **Next** to proceed.

The **Summary** screen displays the information that you entered.

4. Click **Finish** to complete the configuration process.

Adding a member to an SMS cluster

You might need to add a member to an SMS cluster.

About this task

To add a cluster member, take these steps:

- Unconfigure and undeploy the cluster.
- Redeploy and reconfigure the cluster.

For more information, see “smscfg” on page 79.

Procedure

1. Install WebSphere Application Server Network Deployment including appropriate fix packs on the new server and create an appropriate profile.
2. Install WebSphere eXtreme Scale and associated fixes, augmenting the new profile during installation.
3. Federate the new node into the existing cell.
4. Create an WebSphere Application Server instance in the cluster.

SMS pdsmsadmin and pdadmin commands

The **pdsmsadmin** and **pdadmin** command line utilities can be installed as part of the Security Access Manager package.

Use these interfaces to manage access control lists, groups, servers, users, objects, and other resources in your secure domain.

You can also automate certain management functions by writing API scripts that use the **pdadmin** commands, which include an optional delimiter to specify session management server instances.

Chapter 3. Session management server usage

You can administer the session management server and the sessions that it maintains by using the various tools described in "Session management server administration options" on page 1.

Tasks that you can complete by using the Session Management Server console, **pdsmsadmin**, or **pdadmin** include locating, refreshing, and terminating sessions, managing key information for validating external sessions IDs, and generating new keys.

Additionally, you can use the **smscfg –action config** utility to modify the configuration by completing tasks such as:

- Moving replica sets from one session realm to another.
- Add and remove session realms.
- Add and remove replica sets that are not assigned to a session realm.

This chapter details how to complete various common tasks, principally by using the Session Management Server console. For further details on **pdsmsadmin** and **pdadmin** commands, see "SMS pdsmsadmin and pdadmin commands" on page 32. For details of **smscfg –action config** commands, see "smscfg" on page 79.

Logging in and logging out of the Session Management Server console

Access the Security Access Manager Session Management Server console by opening a Web browser and typing the appropriate URL.

About this task

To form the appropriate URL, you must know the settings for the console. For example, the URL you need might be:

```
https://isam.example.com:9043/ibm/console
```

This URL is made up of:

- The name of the host system that runs the console. For example:
  ```
  https://isam.example.com
  ```
- The port number of the console. The port that is used by the Session Management Server console is the same as the port used by the console of the hosting WebSphere Application Server.
- The URL for accessing the console login page. This part of the URL is always the same:
  ```
  /ibm/console
  ```

When you have established the correct URL, you need the administrator user name and password to log in to the console. The name and password were specified during configuration.

Use this procedure to log in to the console:

Procedure

1. Enter the console URL in the address bar of your browser window. For example, for the URL for a system with a host name of `isam.example.com`, using the default port number, enter:

   ```
   https://isam.example.com:9043/ibm/console
   ```
2. Enter the administrator ID and password. For example:
   - User name: `isamadmin`
   - Password as specified when the console was installed.

   The console Welcome panel is displayed.
3. Use the navigation links on the left to view and work with the console tasks.

   Note: Do not use the **Back** button in your browser to move among pages in the console.
4. To log out, click **Logout** in the upper right corner of the panel.

Searching user sessions

You might need to search user sessions.

About this task

All currently active sessions can be listed, or more refined searches of sessions can be made.

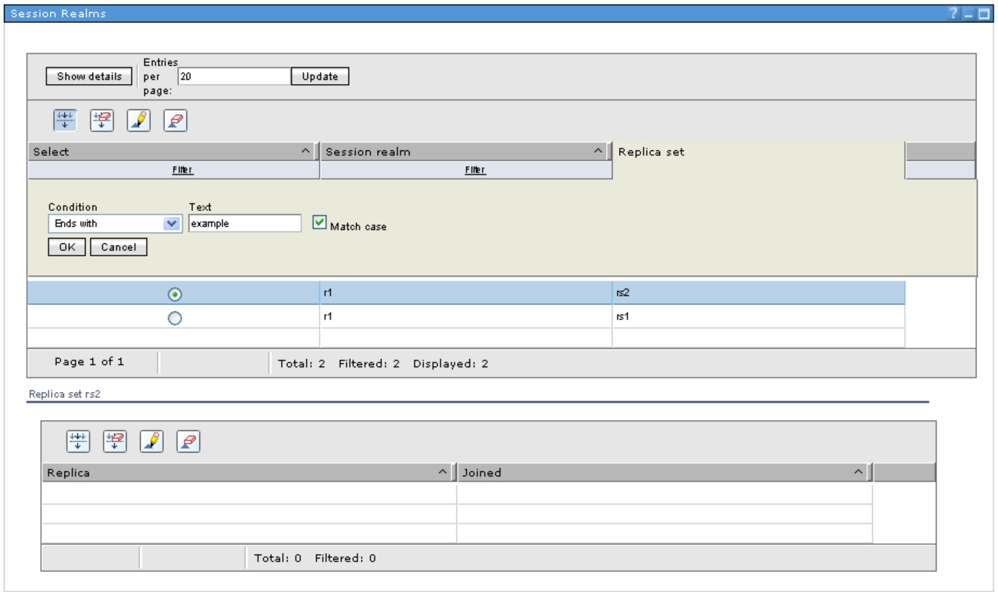
Procedure

1. Select **Search Sessions** from the Session Management Server console main menu.
2. Determine the session management server **instance** you wish to search.

   The following fields are available to restrict your search:
   - The **User ID** field accepts wildcard values.
   - The **Maximum Results** field restricts the number of returned session IDs.
   - The **Session Realm** field must include a session realm value.
3. Click the **Search** button to start your search.

Results

Search results are returned showing the session user ID and the login time for the session. Users with multiple sessions are grouped together. You can select, filter and sort the results using the appropriate icons.



Entries per page: 20

Condition: Ends with

Text: example

☑ Match case

Select	Session realm	Replica set
●	r1	rs2
○	r1	rs1

Page 1 of 1   Total: 2 Filtered: 2 Displayed: 2

Replica set rs2

Replica	Joined

Total: 0 Filtered: 0

Figure 8. Filtered searching of session realms using the Session Management Server console

The **pdsmsadmin** and **pdadmin** command line utilities provide equivalent commands, such as **session list** for listing user sessions. The usage of these commands is described in detail in "SMS pdsmsadmin and pdadmin commands" on page 32.

Ending of user sessions

The ability to terminate user sessions is often useful when, for example, a browser error occurs and the user cannot log back in. After the administrator locates and ends the active session, the user must authenticate again to create a session.

To end one or more user sessions, use the Session Management Server console to search the current sessions, see "Searching user sessions" on page 34 for details. Select the check box corresponding to the user ID you want to end and click **Terminate**.

The **pdsmsadmin** and **pdadmin** command-line utilities provide equivalent commands for ending user sessions by using either **terminate session** or **terminate all_sessions**. The usage of these commands is described in detail in "SMS pdsmsadmin and pdadmin commands" on page 32.

Policies for setting the maximum concurrent sessions

The **policy get** and **policy set** commands in **pdadmin** allow you to display and set maximum concurrent Web session policy. These are standard Security Access Manager **pdadmin** commands, which can be useful for session management purposes but are not available in **pdsmsadmin** or the Session Management Server console.

The following command returns an integer value corresponding to the maximum permitted Web sessions for a user:

```
policy get max-concurrent-web-sessions [-user user_name]
```

The search can be performed for a specific user by employing the **–user** *user_name* option.

```
policy set max-concurrent-web-sessions {unset|number|displace|unlimited}
[-user user_name]
```

This command sets the maximum number of Web sessions the session management server will permit for any one user.

For details of the syntax used with these commands, see the *IBM Security Access Manager for Web: Command Reference*.

Note:

1. When an administrator switches to another user, the new session for the target user will not be subject to concurrent session policy.
2. This functionality is only available if the **session limit policy** option has been enabled during the configuration of the SMS instance.

Commands for displaying session realms and replica sets

An authorized user can monitor session activity, display session realms, list the participating replica sets, list current sessions and search for specific sessions.

The session management server components can be displayed using the Session Management Server console or the command line utilities.

The following commands are available in **pdadmin** and **pdsmsadmin**:

- **realm show**
- **realm list**
- **replica set show**
- **replica set list**

The usage of these commands is described in detail in "SMS pdsmsadmin and pdadmin commands" on page 32.

Management of replica sets and realms

Replica sets and session realms can be managed using the **configuration** menu of the Session Management Server console.

New realms and replica sets can be added by working through the **configuration** dialogs. To add a new session realm or replica set, specify a name that does not already exist. To modify an existing realm or replica set, enter the name of an existing realm or replica set.

Note: Clients which have the same configuration can connect to the same replica set, but clients with different configurations must connect to different replica sets.

Realms and replica sets can be removed by selecting the relevant checkbox and clicking the **Remove Selected** button.

For further information, see Chapter 2, "Configuration," on page 21, in particular the section "Deployment and configuration of more instances" on page 29.

Managing keys

The session management server uses a key to sign session IDs. You can obtain the details of the session signing key from the Session Management Server console.

About this task

This signing key lessens the possibility of a denial of service attack against the session management server. A single key is used across the entire cluster.

The **`pdsmsadmin`** and **`pdadmin`** utilities provide equivalent commands for managing keys using **`key show`** and **`key change`**. These commands are detailed in "SMS pdsmsadmin and pdadmin commands" on page 32.

Procedure

1. From the Session Management Server console menu, select **Key Management**.
2. The Key Management screen is displayed. Use the current instance, or select another session management server instance.

   The screen displays information about the current key. The date and time information is local to the application server that is hosting the WebSphere Application Server.
3. You can force the creation of a new key by clicking **Generate new key now**. You might want to forcibly create a new key when you suspect that the existing key has been compromised.

Session management server statistics

Server statistics for Security Access Manager session management server 7.0 are visible in the WebSphere ISC. To view these statistics, click **monitoring and tuning** > **performance viewer**.

The following statistics can provide a useful overview of session activity:

session lifetime
How long, on average, user sessions are lasting.

session limit enforcements
The number of times users have been denied a login due to concurrent session limits - only useful if there is a concurrent session limit.

session displacements
The number of times users have displaced an existing session to log in - only useful if session displacement is enabled.

active sessions
The number of currently active sessions.

active clients
The number of web security servers currently accessing the SMS.

To examine how sessions are created and deleted over a period of time, you can reset the sessions created, logouts, terminations, idle timeouts, and discarded sessions statistics, record them for a set duration, then plot the results.

Other session management statistics are not particularly useful from a customer perspective.

For complete details about using WebSphere statistics, see the WebSphere documentation.

Login activity tracking

The session management server can record information about the last time a user logged in and the number of failed attempts to login since the last successful login. This information is useful when displayed to users at the time of login. This information alerts users of any potential illegal activity on their account.

A sample JSP file for displaying last login information is available in the following operating system-specific directory:

AIX, Linux, and Solaris operating systems
`/opt/pdsms/etc`

Windows operating systems
`C:\Program Files\Tivoli\PDSMS\etc`

This file can be used as a template for customizing your own display of last login information.

Note: The login information displayed is dependent upon the browser that a user employs to access the system. Therefore, an unsuccessful attempt to login from another browser will not be displayed on the original browser. For example, consider the following activity.

```
1. Login at 12:00
2. Logout at 12:20
3. Login failure at 12:25
4. Login failure at 12:26
5, Login at 12:27
6. Login failure (from another browser) at 12:30
7. Display data via 1st browser
```

The following information would be displayed to the user:

```
Last Login=12:00, 2 login failures since that time, last failure at 12:26.
```

The login history is displayed as it was at the time of login only. Later events are not displayed.

Storage of login activity data

Login activity information is stored in session management server using a JDBC data source. This database is installed at the time of session management server installation. Login activity information is stored using the schema listed in Table 2.

Table 2. Login activity database schema.

Value	Data type	Description
UserName (Primary Key)	String	The unique user name of the user.
UUID	String	UUID for the user name.
nFailures	Integer	A count of the failed logins since the last successful login.
LastLoginFailure	String	A date/time stamp of the last failed login.
LastLoginSuccess	String	A date/time stamp of the last successful login.

A mechanism for reconciling this data with the user registry is not provided. This data schema can be used to develop your own reconciliation capability.

Information about creating the login activity database

The sessions management server uses the generic JDBC interface provided by WebSphere to communicate with the database that is used for storing last login data. Despite JDBC being a common interface, different JDBC implementations provided by different database vendors often behave differently. This particularly pertains to database schema operations.

For details about creating the login activity database, see the **Creating the login history database** section in the *IBM Security Access Manager for Web: Installation Guide*.

If you use a database other than DB2 and you enable the last login data tracking capability of the session management server and then the session management server configuration fails when creating the last login database, you must create the last login database manually and restart the session management server configuration procedure.

The details of the schema required by the session management server for the last login database are provided in Table 2 on page 38 to enable you to manually create this database.

Rules for credential refresh

Where a WebSEAL or Web Plug-in server has been configured for step-up authentication and SMS session storage, a user will be prompted to re-authenticate whenever an administrator refreshes user credentials via the SMS.

This re-authentication occurs because the default SMS credential refresh configuration does not preserve the "AUTHENTICATION_LEVEL" attribute in a user's credential. After their credential is refreshed, the authentication level is reset to zero, so any POPs that require higher authentication levels will result in the user being prompted to login again.

To prevent this re-authentication from occurring, you must update the SMS credential refresh configuration to include a rule that preserves the "AUTHENTICATION_LEVEL" attribute. This configuration can be done using the graphic user interface or command line.

Using the GUI to set rules for credential refresh

You can use the GUI to set rules for credential refresh.

Procedure

1. Source the WebSphere `setupCmdLine.sh` or `setupCmdLine.bat` file to configure your Java environment.
2. Invoke the PDSMS configuration utility: `/opt/pdsms/bin/smscfg -action config` (for UNIX), or `*PDSMS_Install_Dir*/bin/smscfg -action config` (Windows). This launches the graphical configuration utility for the SMS.
3. Enter any required security information to contact the WebSphere server.
4. Once you reach the SMS configuration screens, click **Next** on each screen until you reach the **Specify the credential attribute refresh rules** dialog.

5. Click **Add Rule** to create a new rule.
6. Click on the **Refresh** entry of the new rule and change it to **Preserve**.
7. Click on the * entry of the rule and change it to **authentication_level**.
8. Click **Next** until you reach the summary page.
9. Click **Finish** to start the configuration update.
10. When the configuration update is complete, click **OK** to exit the configuration utility.

Using the command line to set rules for credential refresh

You can use the command line to set rules for credential refresh.

Procedure

1. Create an SMS configuration response file `smsconfig.rsp` that contains any information necessary to contact the WebSphere server, such as the hostname of the WebSphere Application Server or deployment manager, the SOAP port number, and any WebSphere Application Server security information. For example:

   ```
   was_host=wasdm.example.com
   was_port=8880
   was_enable_security=yes
   was_admin_id=wasadmin
   was_admin_pwd=secret123
   trust_store=/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/etc
   /AS01ClientTrustFile.jks
   trust_store_pwd=passw0rd
   ```

2. Update the `smsconfig.rsp` file to include the credential refresh rules you want to specify. Remember to include any existing credential refresh rules. The default credential refresh rule set is `preserve=tagvalue_*`. To add a rule to preserve the **authentication_level** attribute, include these two lines in the `smsconfig.rsp` file:

   ```
   cred_refresh_rule=preserve=authentication_level
   cred_refresh_rule=preserve=tagvalue_*
   ```

3. Source the WebSphere `setupCmdLine.sh` (UNIX) or `setupCmdLine.bat` (Windows) file to configure your Java environment.
4. Change directories to the location of the **smscfg** program: `/opt/pdsms/bin` (UNIX) or *`PDSMS_Install_Dir`*`/bin` (Windows).
5. Invoke the SMS configuration tool with your response file:

   ```
   ./smscfg -action config -interactive no -rsp_file path-to-smsconfig.rsp
   ```

   The configuration tool will update your configuration and restart the DSess application.

Chapter 4. Session management server best practices

This chapter describes best practices for the session management server.

The chapter covers the following topics:

- "Zone rules and zone configuration"
- "Load balancer settings" on page 43
- "SMS sessions and session limit policy" on page 43
- "Java virtual machine heap size" on page 44
- "Session Management Server high availability considerations" on page 45

Zone rules and zone configuration

You can use a combination of zone rules, zone configuration, and the Session Management Server (SMS) client configuration to minimize traffic in an SMS environment.

Zone configuration

A well-planned combination of WebSphere eXtreme Scale zones can minimize the SMS traffic between zone boundaries. When using zones in the SMS, the client configuration must reflect the SMS zone configuration to minimize cross-zone traffic.

Zones can represent computer, building, or data boundaries. Rules determine how partitions are laid out in and between the zones. Use the WebSphere Application Server Network Deployment node group feature to configure the zone for a server.

Zone rule

The SMS includes a default zone rule and populates it with appropriate zone names. The WebSphere Application Server management interface provides these zone names during deployment. The default rule places primary and synchronous replicas in the same zone, and asynchronous replicas in a different zone.

The WebSphere Application Server Network Deployment node groups define zones by prefixing the zone name with `ReplicationZone`. For example, the node group named `ReplicationZoneOne` represents the zone named `One`. Each server can belong to:

- Multiple node groups.
- Only one eXtreme Scale zone. If the deployment process discovers a server that belongs to multiple zones, it displays an invalid configuration error.

WebSphere eXtreme Scale zones

WebSphere eXtreme Scale zones and zone rules control the placement of partitions across the grid. Zones group servers in a particular location. Zone rules define how partitions are placed in and across these zones. Zone-preferred routing permits

WebSphere eXtreme Scale clients to write to WebSphere eXtreme Scale servers in specific zones. Writing to specific zones limits the amount of cross-zone traffic in a grid.

The following diagram shows an example deployment using eXtreme Scale zones.

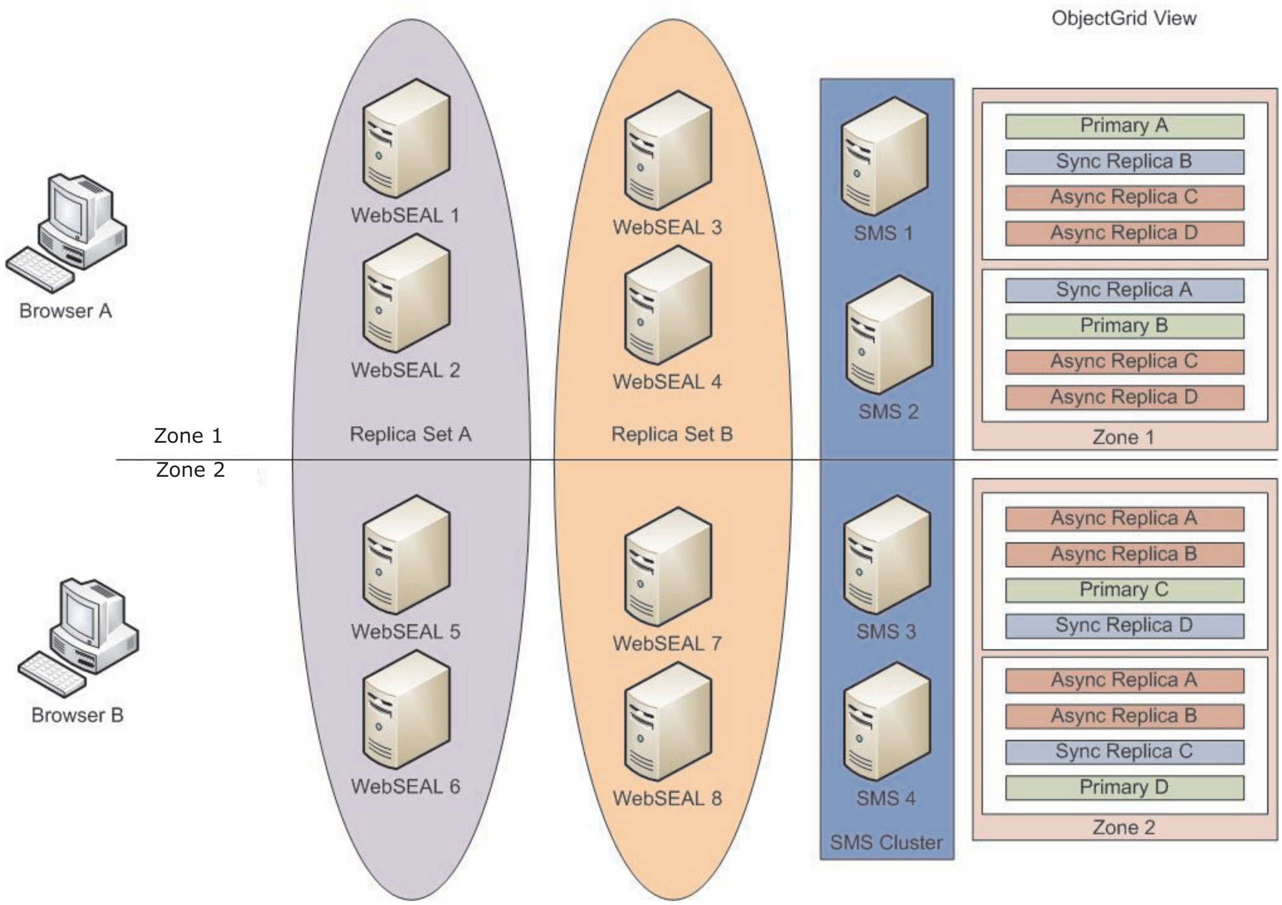


Figure 9. Example deployment using eXtreme Scale Zones

You can find more information about eXtreme Scale zones in the WebSphere eXtreme Scale Version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp

See "Using zones for replica placement" and "Zone-preferred routing" in the *WebSphere eXtreme Scale Product Overview*.

The WebSphere Application Server Information Center also describes how to configure eXtreme Scale zones before deploying the SMS. The Information Center includes instructions for configuring cluster members to be part of a node group that represents a zone. See the "Viewing, adding, and deleting node group members" section in the *Network Deployment (All operating systems), Version 8.0 Guide* in the Websphere Application Server Version 8.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp

Load balancer settings

You can use many different load balancers and numerous algorithms. `Session Stickiness` in the load balancer is the most important setting for an SMS environment. *Stickiness* is the ability of a load balancer to forward all requests from a particular client to the same server during the client session. Use the client IP address to maintain session stickiness.

Use a load balancer to:

- Spread the load equally across a cluster of SMS clients.
- Provide fault tolerance and scalability.

You must set session stickiness for an SMS cluster to function efficiently and effectively. Set the timeout for session stickiness to be equal to the timeout of the client session as defined by `timeout` in the `session` stanza of the configuration file.

SMS sessions and session limit policy

You can enable a session limit policy and set the number of concurrent sessions in the SMS environment.

Session limit policy

The SMS can enforce a session limit policy. This policy can either be on a per user basis, a per realm basis, or enforce session displacement for users. The configuration of the session limit policy occurs in the SMS client (for example, WebSEAL) and is sent to each SMS.

During the SMS configuration, you can disable the session limit policy so that the SMS ignores the limits sent by the client. Disabling the session limit policy at the SMS causes enhanced performance in general operations. However, disabling the session limit policy reduces internal indexing, which reduces performance in administration operations such as session search and terminate.

Number of concurrent sessions

The number of concurrent sessions in the SMS environment dictates how much memory each SMS uses. This value also partially determines the required number of SMS servers. The number of authentications per second required, the session idle timeout, the maximum session lifetime, and the number of logouts determine the number of concurrent sessions.

Example

Consider an environment with four SMS clients and four SMS cluster members. Assume 50 authentications per second with an idle timeout of 30 minutes and a maximum session time of 1 hour. The calculations that follow assume that 80% of the sessions expire and the remaining 20% reach the maximum session lifetime.

50 auth/sec * 60 = 300 auth/minute

300 auth/min * 30 minutes = 9000 sessions created in 30 minutes

After that time, 80% of the sessions become idle = 7200; 1800 remain.

In the next 30 minutes another 9000 are created, making a total of 10800 sessions.

So at any one time, up to 11,000 sessions can be in the SMS.

Session cache size

The sum of the maximum number of sessions that can be held by each SMS client (for example, WebSEAL) determines the maximum concurrent sessions in the SMS. When the client session caches are full, the local cache drops sessions based on a Least Recently Used (LRU) algorithm. Sessions might be removed from the SMS if no other client instances hold a reference to the session.

For example, consider an environment with a maximum of 10,000 sessions in the SMS. To accommodate this number of sessions, set the minimum cache size for each of the four clients to 2500. This setting is not adequate if two clients fail. In this situation, the maximum number of sessions per client is 5000. In this case, set the cache size of each client to 5000 users.

Java virtual machine heap size

In most cases, you must increase the heap size of WebSphere Application servers, including those servers which act as part of the catalog service. These servers can include application servers which run the SMS application, node agents, or the deployment manager. An increase in the Java virtual machine (JVM) heap size is required if many concurrent sessions must be managed by the SMS.

The JVM heap size dictates the maximum amount of memory that can be allocated to a WebSphere Application Server instance. For SMS, it defines the amount of memory that is allocated to the WebSphere eXtreme Scale server, and therefore the session table size.

Note: Servers acting as the catalog service do not maintain SMS session information but they require a larger heap size than the default 256 MB.

Increasing the WebSphere Application Server heap size

You might need to increase the WebSphere Application Server heap size.

Procedure

1. Open the Integrated Solutions Console.
2. On the left-hand side, expand the **Servers** heading and click **Application servers**.
3. Click the name of the WebSphere Application Server you want to modify.
4. Under **Server Infrastructure**, expand the **Java and Process Management** heading and click **Process Definition**.
5. Under **Additional properties**, select **Java Virtual Machine**.
6. In the **Maximum Heap Size** text box, specify the new maximum heap size.
7. Click **OK**.
8. Click **Save**.
9. Restart the application server for the changes to take effect.

Increasing the WebSphere Application Server Deployment Manager heap size

You might need to increase the heap size of the WebSphere Application Server Deployment manager.

Procedure

1. Open the Integrated Solutions Console.
2. On the left-hand side, expand the **System Administration** heading and click **Deployment manager**.
3. Under **Server Infrastructure**, expand the **Java and Process Management** heading and click **Process Definition**.
4. Under **Additional properties**, select **Java Virtual Machine**.
5. In the **Maximum Heap Size** text box, specify the new maximum heap size.
6. Click **OK**.
7. Click **Save**.
8. Restart the deployment manager for the changes to take effect.

Increasing the WebSphere Application Server Node Agent heap size

You might need to increase the WebSphere Application Server Node Agent heap size.

Procedure

1. Open the Integrated Solutions Console.
2. On the left-hand side, expand the **System Administration** heading and click **Deployment manager**.
3. Click the name of the node agent that you want to modify.
4. Under **Server Infrastructure**, expand the **Java and Process Management** heading and click **Process Definition**.
5. Under **Additional properties**, select **Java Virtual Machine**.
6. In the **Maximum Heap Size** text box, specify the new maximum heap size.
7. Click **OK**.
8. Click **Save**.
9. Restart the deployment manager for the changes to take effect.

Object Request Broker configuration

To configure the Object Request Broker properties to suit your environment, see the Orb properties file section in the *WebSphere eXtreme Scale Administration Guide* in the WebSphere eXtreme Scale Version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp

Session Management Server high availability considerations

When deployed, the SMS is a critical Security Access Manager component. Consequently, the SMS must have high availability. Otherwise, failure might cause the client to become unavailable for session operations.

Availability improves if you run the SMS in a clustered environment consisting of at least two cluster members. If all cluster members become unavailable, the session data maintained by the SMS is lost and the SMS is no longer able to service any requests. In this case, the Web security clients (such as WebSEAL) cannot create new sessions.

The availability of the SMS is determined by two separate components in the environment. The first component consists of the actual WebSphere Application Servers running an SMS instance and holding WebSphere eXtreme Scale containers. The second key component is the WebSphere eXtreme Scale catalog service responsible for maintaining routing information for the SMS servers. It is important to consider the high availability of each component as part of an SMS deployment.

Hard failure detection

In a clustered environment, the SMS stores session data such that system operation is not affected by the loss of a single SMS server. You must tune the hard failure detection mechanism in the underlying WebSphere Application Server to optimize failure detection.

Hard failure detection is a means of detecting a physical computer crash, network cable disconnect, or OS panic. WebSphere eXtreme Scale uses a heartbeat detection mechanism to detect hard failure events by using the underlying WebSphere Application Server core group heartbeat feature.

Hard failure detection takes approximately 200 seconds in a default SMS configuration. This value is too large for the SMS to function correctly during a hard failure scenario. A hard failure must be detected in less than 20 seconds for the SMS to function as expected.

Specify the heartbeat interval in milliseconds. You must also specify the number of missed heartbeats. This value indicates how many heartbeats can be missed before a peer JVM is considered failed. The hard failure detection time is approximately the product of the heartbeat interval and the number of missed heartbeats. Specify these properties by using custom properties on the core group in the administrative console.

Consider network performance and reliability to tune these settings appropriately for the specific environment. When these settings are too aggressive, false failures are detected. However, if these settings are not aggressive enough, failures are not detected early enough for the system to recover in a suitable time frame.

WebSphere Application Server Network Deployment versions 7.0 and 8.0 provide the following two core group settings:

- Heartbeat transmission period.
- Heartbeat timeout period.

These settings can be adjusted to increase or decrease failover detection.

For more information about heartbeat detection in WebSphere Application Server, see the Core group heartbeat configuration topic under the Configuring failover detection section in the *WebSphere eXtreme Scale Administration Guide* in the WebSphere eXtreme Scale Version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp

eXtreme Scale container JVMs

The eXtreme Scale container JVMs are the virtual machines that hold the SMS session data. The WebSphere Application Servers in the SMS cluster are also known as the eXtreme Scale Container JVMs.

Container failure

When a container fails, any primaries that are held in that session management server are promoted elsewhere in the SMS cluster.

The SMS can withstand multiple container JVM failures before data loss occurs, if at least one set of primaries for all the maps remains available.

The WebSphere eXtreme Scale catalog service promotes and demotes containers according to the deployment parameters defined in the SMS. It is important however that container JVMs continue to maintain a connection to the catalog service.

If a container JVM loses contact with the catalog service and then regains contact, the catalog service detects the failure. Any containers that are held by the container JVM are dropped and placed on other servers. Under these conditions, the container JVM continues to act as an eXtreme Scale client, but you must manually restart the JVM to hold the containers again. This behavior is designed to ensure consistency of the data across the cluster.

eXtreme Scale Catalog service

A catalog service is the grid of catalog servers you are using. You can run catalog servers inside WebSphere Application Server JVM processes or inside stand-alone JVM processes. These servers retain topology information for all the containers in your eXtreme Scale environment, which is the SMS deployment environment in this case.

The catalog service controls balancing and routing for all clients and manages the promotion and demotion of containers from primaries to replicas. For eXtreme Scale to operate for the SMS, you must cluster the catalog servers into a grid for high availability.

When the catalog service starts, it selects a master catalog server that is responsible for holding the master copy of all data in the catalog service. The master catalog server accepts Internet Inter-ORB Protocol (IIOP) heartbeats and handles system data changes in response to any catalog service or container changes.

When clients contact any of the catalog servers, the routing table for the catalog server grid is propagated to the clients. This propagation occurs through the Common Object Request Broker Architecture (CORBA) service context.

To ensure high availability, configure at least two catalog servers into a catalog service cluster. If your configuration has zones, you can configure one catalog server per zone.

When an eXtreme Scale server and container contacts one of the catalog servers, the routing table for the catalog server grid is also propagated to the eXtreme Scale server and container. This propagation occurs through the CORBA service context. If the contacted catalog server is not currently the master catalog server, the request is automatically rerouted to the current master catalog server. The routing table for the catalog server is also updated.

Note: A catalog server grid and the container server grid are different. The catalog server grid is for high availability of the eXtreme Scale system data. The container grid is meant for high availability, scalability, and workload management of the SMS application. Consequently, two different routing tables exist:

- The routing table for the catalog server grid, and
- The routing table for the server grid shards.

Catalog servers which run inside WebSphere Application Server JVM processes

You can configure the WebSphere Application Server instances to run WebSphere eXtreme Scale catalog servers.

You can configure the catalog service to run in any process in the WebSphere cell. A single-server catalog service is acceptable for development environments. For a production environment, use a catalog service grid with multiple catalog servers.

For WebSphere Application Server Network Deployment, the catalog service runs in the deployment manager process automatically. However, you can configure the catalog service to run in one or more application server processes.

The WebSphere eXtreme Scale catalog service configuration is defined by using the `catalog.services.cluster` custom property in the WebSphere cell. To run the catalog service inside a WebSphere Application Server, use the following format for the value of this property:

```
<serverName>:<hostname>:<clientPort>:<peerPort>:<listenerPort>
[,<serverName>:<hostname>:<clientPort>:<peerPort>:<listenerPort> ...]
```

where

serverName
Specifies the fully qualified name of the WebSphere process, such as the cellName, nodeName, and serverName of the server that hosts the catalog service.

hostname
Specifies the name of the hosting server.

clientPort
Specifies the port that is used for peer catalog grid communication.

peerPort
Specifies the port that is used for peer catalog grid communication. The port can be anything that you choose in a WebSphere Application Server environment.

listenerPort
The listenerPort must match the BOOTSTRAP_ADDRESS value that is defined in the WebSphere server configuration.

Example:

```
sms1Cell01\sms1CellManager01\dmgr:sms1.amtest.gc.au.ibm.com:6600:6601:9809,
sms1Cell01\sms4Node02\catalogServer1:sms4.amtest.gc.au.ibm.com:6602:6603:2809
```

For more information, see "Starting the catalog service process in a WebSphere Application Server environment" in the *WebSphere eXtreme Scale Administration Guide*. The guide is in the WebSphere eXtreme Scale Version 7.0 Information Center:

http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp

Catalog servers which runs inside stand-alone JVMs

Depending on the architecture of an SMS deployment, you can choose to run the catalog servers inside stand-alone JVMs rather than inside WebSphere Application Servers.

You must define a WebSphere Application Server cell custom property that is called `catalog.services.cluster` so that the WebSphere eXtreme Scale container servers can contact the catalog service. For catalog servers which run inside stand-alone JVMs, use the following format to specify the value of this property:

```
<serverName>:<hostname>:<clientPort>:<peerPort>[,<serverName>:<hostname>:
<clientPort>:<peerPort>...]
```

where

serverName
: Specifies a name to identify the process to run.

hostname
: Specifies the host name of the computer where the server run.

clientPort
: Specifies the port that you are using for peer catalog grid communication.

peerPort
: Specifies the port that you are using for peer catalog grid communication.

Example:

```
cat1:sms-multicell01.vam.gc.au.ibm.com:6602:6603:2809,cat2
:sms-multicell02.vam.gc.au.ibm.com:6602:6603:2809
```

The catalog service can run in a single process or can include multiple catalog servers to form the catalog server grid. For high availability, a production environment requires a catalog server grid. Whether the catalog service is placed in a grid, or a single process, use the `startOgServer` script to start the service.

For more information, see the Starting the catalog service in a stand-alone environments section in the *IBM WebSphere eXtreme Scale Administration Guide*. The guide is in the WebSphere eXtreme Scale, version 7.0, information center:

http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp

Catalog service failure

The catalog service grid is a WebSphere eXtreme Scale grid that uses the core grouping mechanism in the same way as the container failure process. The catalog service grid uses a peer election process to define the primary shard instead of the catalog service algorithm that is used for containers.

Catalog service members must be contained in the same core group. However, the catalog service core group do not have to be the same as the core group where the SMS servers are located.

The catalog service uses replication to make itself fault-tolerant. If a catalog service process fails, restart the service to restore the system to the level of availability that you want. If all the processes that are hosting the catalog service fail, WebSphere eXtreme Scale loses critical data. This failure results in a required restart of all the containers.

The catalog service can run on many processes so this failure is an unlikely event. However, a failure might occur if you are running all the processes:

- On a single box
- In a single blade chassis
- From a single network switch
-

Remove common failure modes from boxes that are hosting the catalog service to reduce the possibility of failure.

Catalog service quorum behavior

Normally, the members of the catalog service have full connectivity. The catalog service grid is a static set of JVMs. WebSphere eXtreme Scale expects all members of the catalog service to be always online. The catalog service responds only to container events while the catalog service has quorum.

If the catalog service loses quorum, it waits for quorum to be reestablished. When the catalog service does not have quorum, it ignores events such as failures from container servers.

The following message indicates that quorum is lost. Look for this message in your catalog service logs.

```
CWOBJ1254W: The catalog service is waiting for quorum.
```

WebSphere eXtreme Scale expects to lose quorum in the following situations:

- Catalog service JVM member failure
- Network brownout
- Loss of connectivity between zones

In terms of the SMS, lost quorum in the catalog service does not affect service. The Web security server environment continues to operate during quorum loss provided no container server loss occurs.

Quorum loss from JVM failure

A catalog server that fails due to a crash causes quorum to be lost. In this event, manually override quorum as quickly as possible. The failed catalog service cannot rejoin the grid until you manually override quorum.

For further information about overriding quorum, see “Overriding quorum” on page 53.

Quorum loss from network brownout

WebSphere eXtreme Scale handles the possibility of brownouts. A brownout is temporary loss of connectivity between nodes. This loss of connectivity is generally transient and brownouts typically clear in a matter of seconds or minutes.

WebSphere eXtreme Scale tries to maintain normal operation during a brownout period. However, WebSphere eXtreme Scale regards a brownout as a single failure event. The failure is expected to be fixed and then normal operation resumes with no WebSphere eXtreme Scale actions necessary.

During a brownout, if quorum is lost in the catalog service and SMS servers are disconnected from one another, sessions can no longer be stored. After the brownout clears, normal operation resumes.

Catalog service JVM cycling

If you stop a catalog server by stopping the JVM that it is running in, then the quorum drops to one less server. This means that the remaining servers still have quorum.

Restarting the catalog server restores quorum to the previous number.

Consequences of lost quorum

If a container JVM fails while quorum is lost in the catalog service, recovery does not take place until the brownout or blackout ends. To recover from a blackout, override the quorum command manually.

WebSphere eXtreme Scale considers a quorum loss event and a container failure as a double failure, which is a rare event. Until quorum is restored and normal recovery can take place, applications such as the SMS might lose write access to data which are stored on the failed JVM. In the context of a Security Access Manager Web security server environment, this double failure means that sessions cannot be stored in the SMS.

Similarly, if you attempt to start a container during a quorum loss event, the container does not start. If you try to start an SMS server during quorum loss, the SMS server does not start properly.

Full client connectivity is allowed during quorum loss. If no container failures or connectivity issues happen during the quorum loss event, then clients can still fully interact with the container servers. As a result, the SMS can store and access sessions during quorum loss provided no SMS servers fail, or become noncommunicable, at the same time.

If a brownout occurs, then some clients might not have access to primary or replica copies of the data until the brownout clears. Under some brownout conditions, the SMS can continue the session operations, depending on the primaries available to the particular server at the time.

Quorum recovery

If quorum is lost and reestablished, a recovery protocol is executed by the catalog service. When the quorum loss event occurs, all liveness checking for core groups is suspended and failure reports are also ignored. When quorum returns, the catalog service performs a liveness check of all core groups to immediately determine their membership.

Any shards previously hosted on container JVMs that reportedly failed are recovered. If primary shards were lost then surviving replicas are promoted to primaries. If replica shards were lost, then additional replicas are created on the survivors. For the SMS, this recovery process means that no session data is lost. However, a significant number of SMS server failures might cause data loss.

Failure scenarios

This section describes scenarios where significant failures lead to loss of service for the SMS and the Web security server environment. The following diagram illustrates an example SMS deployment.

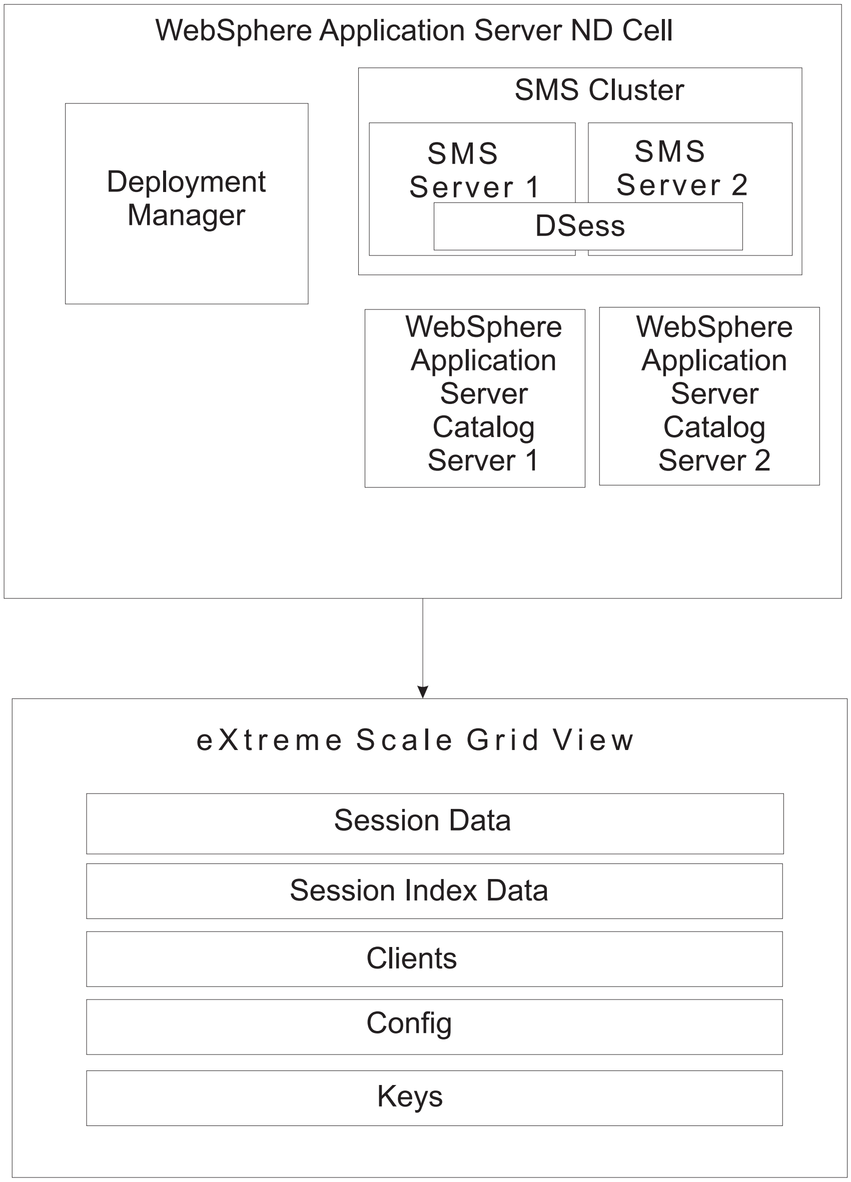


Figure 10. Session Management Server Deployment

Multiple container loss

Multiple container failures might cause loss of data held in the SMS. Container failures can lead to a situation where the SMS cannot store sessions. The likelihood of losing enough containers to impact service is minimal due to the underlying replication and the placement of synchronous and asynchronous replicas that eXtreme Scale provides.

An example of this failure is when both 'SMS Server 1' and 'SMS Server 2' fail. See Figure 10 on page 52.

To mitigate the risk of multiple container failures causing service loss, add additional SMS servers to the cluster. During configuration, you can calculate the number of containers needed based on the size of the cluster. Additional servers in the environment result in the creation of more containers, thus reducing the risk of critical data loss.

Loss of all catalog servers

If all catalog servers fail, the SMS continues to function. However, the SMS has limited ability to cope with container failures. When the catalog service is restored after being offline, the SMS JVMs cannot hold containers. Consequently the SMS cannot store sessions. It is important that catalog servers are on separate hardware from container servers. This separation prevents the hardware failure of a specific computer being a single point of failure.

An example of this failure is when both 'Catalog server 1' and 'Catalog server 2' fail. See Figure 10 on page 52.

Note: The system can continue to operate in this mode. However, when the catalog service becomes available again, there is no guarantee that data is not lost or that any SMS servers hold primaries until they are restarted.

Cluster the catalog service to reduce the risk of losing the entire catalog service.

Container and catalog server loss

While a container server and catalog server are both lost, the SMS cannot service requests. It is important that this failure is either mitigated or recovered from quickly to ensure continual service to the Security Access Manager Web Security Servers.

An example of this failure is when both SMS Server 1 and Catalog server 1 fail. See Figure 10 on page 52.

To mitigate this risk, configure separate nodes for each SMS server and catalog service, and ensure that redundant links exist between nodes.

Recovery procedures

Manual intervention is required to recover from a brownout or a blackout where a catalog service member and a container JVM are lost. This process involves overriding quorum on one side of the brownout or blackout. After the brownout or blackout clears and service resumes, restart the SMS and catalog servers located on the other side of the brownout or blackout.

Overriding quorum

You might need to enable quorum within the catalog service cluster in a WebSphere Application Server environment.

About this task

Quorum is an important concept in an eXtreme Scale environment. When quorum is established, the grid can detect and continue functioning after network brownouts.

For more information about overriding quorum in a clustered catalog service, see "Catalog server quorums" in the *WebSphere eXtreme Scale Product Overview*, located in the WebSphere eXtreme Scale Version 7.0 Information Center:

> http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r0/index.jsp

Procedure

- To enable quorum within the catalog service cluster in a WebSphere Application Server environment:
  1. Create a file `objectGridServer.properties` in the `<WAS_HOME>\profiles\<profile>\properties` directory of each clustered catalog service member.
  2. Specify the following entry in this file:

     ```
     enableQuorum=true
     ```
- To enable quorum in the catalog service cluster in a stand-alone JVM environment, you can do *either* of the following:
  - Pass the `-quorum enabled` flag on the `startOgServer` command.
  - Add the `enableQuorum=true` property in the property file passed in to the `startOgServer` command.

  All the catalog servers must have the same quorum setting.

  If a hard failure leads to the loss of a catalog service member and an SMS node, you must manually override quorum to reestablish service. If the loss is due to a network brownout, quorum is reestablished without any manual intervention when the brownout clears. You must manually intervene if the brownout or loss is permanent.

  You can use the `xsadmin` command-line tool to override quorum. This process enables the catalog service to promote eXtreme Scale replicas to primaries and enables the SMS to become fully functional again. The command to override quorum is as follows.

  On a catalog server that is not the Deployment Manager:

  ```
  > xsadmin.sh -ch <cathost> -p <port> -overridequorum
  ```

  where

  cathost The host of the catalog server where quorum is to be over-ridden.

  port The port of the catalog server (typically 9809 in a WebSphere Application Server Network Deployment environment).

  On a catalog server that is the Deployment Manager:

  ```
  > xsadmin.sh -dmgr -overridequorum
  ```

Appendix A. SMS pdsmsadmin and pdadmin commands

The **pdsmsadmin** and **pdadmin** command line utilities can be installed as part of the Security Access Manager package.

Use these interfaces to manage access control lists, groups, servers, users, objects, and other resources in your secure domain.

You can also automate certain management functions by writing API scripts that use the **pdadmin** commands, which include an optional delimiter to specify session management server instances.

How to read syntax statements

The reference documentation uses the following special characters to define syntax:

[ ] Identifies optional syntax. Options not enclosed in brackets are required.

... Indicates that you can specify multiple values for the previous option.

| Indicates mutually exclusive information. You can use the option to the left of the separator or the option to the right of the separator. You cannot use both options in a single use of the command.

{ } Delimits a set of mutually exclusive options when one of the options is required. If the options are optional, they are enclosed in brackets ([ ]).

\ Indicates that the command line wraps to the next line. It is a continuation character.

The options for each command or utility are listed alphabetically in the Options section or Parameters section, respectively. When the order of the options or parameters must be used in a specific order, this order is shown in the syntax statements.

login

Logs user in to **pdsmsadmin** or **pdadmin** command line. If a password is not provided, you will be prompted for it.

Syntax

PDSMSADMIN
login *username* [*password*]

PDADMIN
login -a *username* **-p** [*password*]

Options

user Specifies the user name.

password
Specifies the user password.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **`pdsmsadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

The following example logs a user named *user1* with a password of *passw0rd* in to the **pdsmsadmin** command line:

```
pdsmsadmin> login user1 passw0rd
```

See also

"trace get" on page 71

set instance

Sets the current instance, allowing you to swap from one instance to another to perform administrative tasks.

Syntax

PDSMSADMIN
set instance *instance*

PDADMIN
Not available: the [*.instance*] delimiter can optionally be specified as part of **pdadmin** commands. If not specified, the first available instance is used.

Options

instance
Specifies the name of the server instance to be set.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **`pdsmsadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

The following example sets *instance1* as the current instance:

```
pdsmsadmin> set instance instance1
```

See also

"trace get" on page 71

instances list

Lists all available instances.

Syntax

PDSMSADMIN
instances list

PDADMIN
instances list

Options

NIL No parameters required.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **`pdsmsadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

The following example lists all available instances:

```
pdsmsadmin> instances list
```

See also

"trace get" on page 71

server list

Lists all registered Security Access Manager servers.

Requires authentication (administrator ID and password) to use this command.

Syntax

PDSMSADMIN
Not available.

PDADMIN
`server list`

Description

Lists all registered Security Access Manager servers. The name of the server for all server commands, except for the **`server list`** command, must be entered in the exact format as it is displayed in the output of this command.

Options

None.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

The following example lists all registered servers if the Security Access Manager component is the authorization server:

```
pdadmin> server list
```

Output is similar to:

```
ivacld-topserver
ivacld-server2
ivacld-server3
ivacld-server4
```

key change

Forces the creation of a new session management key.

You might want to forcibly create a new key when you suspect that the existing key was compromised.

Syntax

PDSMSADMIN
key change

PDADMIN
server task *server_name–host_name* **sms**[.*instance*] **key change**

Options

server_name–host_name
Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be `default-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-cruz.dallas.ibm.com`.

If there are multiple configured server instances on the same machine, for example, the host `cruz.dallas.ibm.com`, and the configured name of the WebSEAL server instance is `webseal2-webseald`, the *server_name* would be `webseal2-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server instance would be `webseal2-webseald-cruz.dallas.ibm.com`.

The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0
The command completed successfully.

1
The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

Examples

The following example forcibly creates a new session management key for the `abc.ibm.com` server:

```
pdsmsadmin> key change
```

```
pdadmin> server task default-webseald-abc.ibm.com key change
```

See also

"server list" on page 57
"key show"

key show

Lists detailed information about the current session management key.

Syntax

PDSMSADMIN
key show

PDADMIN
server task *server_name*–*host_name* **sms**[.*instance*] **key show**

Options

server_name–*host_name*
Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be `default-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-cruz.dallas.ibm.com`.

If there are multiple configured server instances on the same machine, for example, the host `cruz.dallas.ibm.com`, and the configured name of the WebSEAL server instance is `webseal2-webseald`, the *server_name* would be `webseal2-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server instance would be `webseal2-webseald-cruz.dallas.ibm.com`.

The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0
The command completed successfully.

1
The command failed. When a command fails, the **`pdsmsadmin`** or **`pdadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example returns detailed information about the current session management key for the `abc.ibm.com` server:

```
pdsmsadmin> key show
```

```
pdadmin> server task default-webseald-abc.ibm.com sms key show
```

Output is similar to:

```
ID: 1
Created: 2004-03-03-09:00:03
Expires: 2004-09-03-09:00:03
```

See also

"server list" on page 57
"key change" on page 58

realm list

Lists all session management realms in the domain.

Syntax

PDSMSADMIN
realm list

PDADMIN
`server task` *server_name*–*host_name* **sms**[.*instance*] **realm list**

Options

server_name–*host_name*
Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **`server list`** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0
The command completed successfully.

1
The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes®

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example lists the realms for the abc.ibm.com server:

```
pdsmsadmin> realm list
```

```
pdadmin> server task default-webseald-abc.ibm.com sms realm list
```

See also

"server list" on page 57
"realm show"
"replica set list" on page 65
"replica set show" on page 66

realm show

Lists all replica sets in the specified session management realm.

Syntax

PDSMSADMIN
realm show *realm_name*

PDADMIN
server task *server_name–host_name* **sms**[.*instance*] **realm show** *realm_name*

Options

realm_name
Specifies the name of the realm. When you specify a realm, the output contains only those replica sets in that realm.

server_name–*host_name*

Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **`server list`** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be `default-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-cruz.dallas.ibm.com`.

If there are multiple configured server instances on the same machine, for example, the host `cruz.dallas.ibm.com`, and the configured name of the WebSEAL server instance is `webseal2-webseald`, the *server_name* would be `webseal2-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server instance would be `webseal2-webseald-cruz.dallas.ibm.com`.

The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **`pdsmsadmin`** or **`pdadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example returns the replica sets in the `ibm.com` realm of the `abc.ibm.com` server:

```
pdsmsadmin> realm show ibm.com
```

```
pdadmin> server task default-webseald-abc.ibm.com sms realm show ibm.com
```

See also

"server list" on page 57
"realm list" on page 60
"replica set list" on page 65
"replica set show" on page 66

session refresh all_sessions

Refreshes the credential for sessions for a specific user.

Syntax

PDSMSADMIN

session refresh all_sessions *user_name* **–realm** *realm_name*

PDADMIN

server task *server_name*–*host_name* **sms**[.*instance*] **session refresh all_sessions** *user_name* **–realm** *realm_name*

Options

–realm *realm_name*

Specifies that name of the realm. Only sessions that belong to the specified realm will have credentials refreshed.

server_name–*host_name*

Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host cruz.dallas.ibm.com is default, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

user_name

Refreshes the credential for all sessions that are associated with the specified user. Examples of user names are dlucas, sec_master, and "Mary Jones".

[.*instance*]

This delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0
The command completed successfully.

1
The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example refreshes all sessions for user johnq in the ibm.com realm:

```
pdsmsadmin> session refresh all_sessions johnq -realm ibm.com
```

```
pdadmin> server task default-webseald-cruz sms session refresh all_sessions johnq
-realm ibm.com
```

See also

"session terminate session" on page 70
"session terminate all_sessions" on page 68

session refresh session

Refreshes the credential for a session.

Syntax

PDSMSADMIN
: **session refresh session** *session_id* **–realm** *realm_name*

PDADMIN
: `server task` *server_name–host_name* **sms**[.*instance*] **session refresh session** *session_id* **–realm** *realm_name*

Options

–realm *realm_name*
: Specifies that name of the realm. Only sessions that belong to the specified realm will have credentials refreshed.

server_name–host_name
: Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the `server list` command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be `default-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-cruz.dallas.ibm.com`.

If there are multiple configured server instances on the same machine, for example, the host `cruz.dallas.ibm.com`, and the configured name of the WebSEAL server instance is `webseal2-webseald`, the *server_name* would be `webseal2-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server instance would be `webseal2-webseald-cruz.dallas.ibm.com`.

The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

session_id
: Specifies the identifier for the session to refresh.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **`pdsmsadmin`** or **`pdadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example refreshes session 678 in the ibm.com realm:

```
pdsmsadmin> session refresh session 678 -realm ibm.com
```

```
pdadmin> server task default-webseald-cruz sms session refresh session 678
-realm ibm.com
```

See also

"session terminate session" on page 70
"session terminate all_sessions" on page 68

replica set list

Lists all session management replica sets in the domain.

Syntax

PDSMSADMIN
: **replica set list** [**–realm** *realm_name*]

PDADMIN
: **server task** *server_name–host_name* **sms**[*.instance*] **replica set list** [**–realm** *realm_name*]

Options

–realm *realm_name*
: Indicates that the returned list of replica sets is limited to those replica sets in the specified realm.

server_name–host_name
: Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host cruz.dallas.ibm.com is default, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

The [*.instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin**

command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example lists the replica sets in the `ibm` realm of the `abc.ibm.com` server:

```
pdsmsadmin> replica set list -realm ibm
```

```
pdadmin> server task default-webseald-abc.ibm.com sms replica set list -realm ibm
```

See also

"server list" on page 57
"realm list" on page 60
"realm show" on page 61
"replica set show"

replica set show

Lists all session management replicas in the specified replica set with the time and date that each joined the realm.

Syntax

PDSMSADMIN
: **replica set show** *replica_set_name*

PDADMIN
: **`server task`** *server_name–host_name* **sms**[.*instance*] **replica set show** *replica_set_name*

Options

replica_set_name
: Specifies the name of the replica set.

server_name–host_name
: Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **`server list`** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be `default-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-cruz.dallas.ibm.com`.

If there are multiple configured server instances on the same machine, for example, the host `cruz.dallas.ibm.com`, and the configured name of the WebSEAL server instance is `webseal2-webseald`, the *server_name* would be `webseal2-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server instance would be `webseal2-webseald-cruz.dallas.ibm.com`.

The [*.instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0
The command completed successfully.

1
The command failed. When a command fails, the **`pdsmsadmin`** or **`pdadmin`** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example returns details about the `ibm.com` replica set of the `abc.ibm.com` server:

```
pdsmsadmin> replica set show ibm.com
```

```
pdadmin> server task default-webseald-abc.ibm.com sms replica set show ibm.com
```

See also

"server list" on page 57
"realm list" on page 60
"realm show" on page 61
"replica set list" on page 65

session list

Lists all session management sessions.

Syntax

PDSMSADMIN
session list –realm *realm_name pattern maximum_return*

PDADMIN
`server task` *server_name–host_name* **sms**[*.instance*] **session list –realm** *realm_name pattern maximum_return*

Options

–realm *realm_name*
Specifies the name of the session management realm.

server_name–host_name
Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **`server list`** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

maximum_return
Specifies the maximum number of sessions to return. When there are more matches than designated by this option, the output contains the number of matches.

pattern Specifies the pattern for returning user names. The pattern can include a combination of wild card and string constant characters. The pattern is case-sensitive. For example, you can specify *luca* as the pattern to find all users that contain the substring luca in the user name.

[*.instance*]

This delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example (entered as one line) lists the user sessions in the ibm.com realm of the abc.ibm.com server for users that contains the string ons and limits the number of matches to 100:

```
pdsmsadmin> session list -realm ibm.com *ons* 100
```

```
pdadmin> server task default-webseald-abc.ibm.com
sms session list -realm ibm.com *ons* 100
```

See also

"server list" on page 57
"realm list" on page 60
"realm show" on page 61
"replica set show" on page 66

session terminate all_sessions

Terminates all user sessions for a specific user.

Syntax

PDSMSADMIN

session terminate all_sessions *user_id* **–realm** *realm_name*

PDADMIN

server task *server_name–host_name* **sms**[*.instance*] **session terminate all_sessions** *user_id* **–realm** *realm_name*

Options

–realm *realm_name*

Specifies that name of the session management realm.

server_name–host_name

Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host cruz.dallas.ibm.com is default, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

The [*.instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

user_id Specifies the name of the user. Examples of user names are dlucas, sec_master, and "Mary Jones".

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example terminates all sessions for the dlucas user in the ibm.com realm of the default-webseald-cruz WebSEAL server:

```
pdsmsadmin> session terminate all_sessions dlucas -realm ibm.com
```

```
pdadmin> server task default-webseald-cruz sms session terminate \
 all_sessions dlucas -realm ibm.com
```

See also

"session refresh session" on page 64
"session refresh all_sessions" on page 62
"session terminate session"

session terminate session

Terminates a user session using a session ID.

Syntax

PDSMSADMIN

session terminate session *session_id* **–realm** *realm_name*

PDADMIN

server task *server_name–host_name* **sms**[*.instance*] **session terminate session** *session_id* **–realm** *realm_name*

Options

server_name–host_name

Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host cruz.dallas.ibm.com is default, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

The [*.instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

session_id

Specifies the ID of a user session.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example terminates session 678 in the ibm.com realm of the default-webseald-cruz WebSEAL server:

```
pdsmsadmin> session terminate session 678 -realm ibm.com
```

```
pdadmin> server task default-webseald-cruz sms session terminate \
session 678 -realm ibm.com
```

See also

"session refresh all_sessions" on page 62
"session terminate all_sessions" on page 68

trace get

Displays the trace level for the session management server. You can use the Session Management Server console to configure the WebSphere tracing facility, which provides more fine grained control.

Syntax

PDSMSADMIN
: **trace get**

PDADMIN
: **server task** *server_name*–*host_name* **sms**[.*instance*] **trace get**

Options

server_name–*host_name*
: Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

  For example, if the configured name of a single WebSEAL server on host cruz.dallas.ibm.com is default, the *server_name* would be default-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server would be default-webseald-cruz.dallas.ibm.com.

  If there are multiple configured server instances on the same machine, for example, the host cruz.dallas.ibm.com, and the configured name of the WebSEAL server instance is webseal2-webseald, the *server_name* would be webseal2-webseald and the *host_name* would be cruz.dallas.ibm.com. For this example, the name of the server instance would be webseal2-webseald-cruz.dallas.ibm.com.

  The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin**

command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example returns the tracing level for the `ivacld-cruz` authorization server:

```
pdsmsadmin> trace get
```

```
pdadmin> server task ivacld-cruz.dallas.ibm.com sms trace get
```

See also

“trace set”

trace set

Sets the trace level for the session management server. You can use the Session Management Server console to configure the WebSphere tracing facility, which provides more fine grained control.

Syntax

PDSMSADMIN
trace set *level*

PDADMIN
server task *server_name*–*host_name* **sms**[.*instance*] **trace set** *level*

Options

level Specifies the level of tracing. A valid setting is an integer between `0` and `3`, with `3` being the most detailed level of trace.

server_name–*host_name*
Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **server list** command.

For example, if the configured name of a single WebSEAL server on host `cruz.dallas.ibm.com` is `default`, the *server_name* would be `default-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-cruz.dallas.ibm.com`.

If there are multiple configured server instances on the same machine, for example, the host `cruz.dallas.ibm.com`, and the configured name of the WebSEAL server instance is `webseal2-webseald`, the *server_name* would be `webseal2-webseald` and the *host_name* would be `cruz.dallas.ibm.com`. For this example, the name of the server instance would be `webseal2-webseald-cruz.dallas.ibm.com`.

The [.*instance*] delimiter is optional in **pdadmin**. If not specified, the first available instance is used.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdsmsadmin** or **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Notes

The **pdadmin** command is available only when the session management command line extensions are installed to a hosting authorization server.

The following example sets the tracing level to 1 on the ivacld-cruz authorization server:

```
pdsmsadmin> trace set 1
pdadmin> server task ivacld-cruz.dallas.ibm.com sms trace set 1
```

See also

“trace get” on page 71

Appendix B. SMS utilities

This section describes utilities for the session management server.

pdsmsclicfg

Configures the command-line administration utility for the session management server.

Syntax

```
pdsmsclicfg –action config [–rspfile response_file] [–interactive {yes|no}]
[–sam_integration {yes|no}] [–aznapi_app_config_file path_name]
[–webservice_location host:port[,host:port...]] [–instances name1,name2]
[-ssl_enable {yes|no}] [–sslkeyfile path] [–sslkeyfile_stash path]
[–sslkeyfile_label label]
```

```
pdsmsclicfg –action unconfig
```

```
pdsmsclicfg –action name
```

```
pdsmsclicfg –action version
```

```
pdsmsclicfg –action upgrade
```

Description

The **pdsmsclicfg** utility configures or unconfigures the session management server command-line administration utility. A log of the configuration progress is written to the `msg_pdsmsclicfg.log` log file. The log file is in the:

- `/var/pdsms/log` directory on AIX, Linux, and Solaris operating systems.
- *`installation_directory`*`\log` directory on Windows operating systems.

This utility can be run in one of the following ways:

- Interactively – the user is prompted to provide configuration information.
- Silently – the utility accepts input from a response file or the command line.

Integration with Security Access Manager can be enabled during configuration. The program prompts the user to specify the path to the configuration file for a configured **aznapi** application. The program prompts the user to specify the location of the web service. The location of the web service is defined by a host name and port that are separated by a semicolon.

The user can specify multiple locations, when each location is separated by a comma. If this web service uses a secure connection, the program prompts the user for the SSL options. You must also specify the session management server instance.

The configuration information is saved to `/opt/pdsms/etc/pdsmsclicfg.conf`. The presence of this configuration file is used to determine the configuration status of the utility.

The command-line executable program on Windows is `pdsmsclicfg-cl.exe`.

Parameters

–action {config|unconfig|upgrade|name|version}
Specifies an action that is one of the following values:

config Configures the command-line administration utility.

unconfig
Fully unconfigures the command-line administration utility. No other parameters are required.

name Displays the translated "Session Management Command Line" name. No other options are required.

upgrade
Configures an upgrade from a previous version.

version
Displays the version number for the currently installed SMS CLI package.

–rspfile *response_file*
Specifies the fully qualified path and file name of the response file to use during silent configuration. A response file can be used for configuration. There is no default response file name. The response file contains stanzas and *key=value* pairs. For information about using response files, see the "Using response files" appendix in the *IBM Security Access Manager for Web Command Reference*. (Optional)

–interactive {yes|no}
Indicates whether the configuration is interactive. The default value is `yes`. (Optional)

–sam_integration {yes|no}
Specifies whether integration with the Security Access Manager administration framework is required. The default value is `no`. (Optional)

–aznapi_app_config_file *path_name*
Specifies the fully qualified name of the configuration file for the hosting authorization server. Only required if Security Access Manager integration is enabled. (Optional)

–webservice_location *host*:*port*
Specifies the location of the session management server Administration web service. The location is the name of the hosting server and the port on which the web service is located. Multiple locations can be specified. When you specify multiple locations, separate the locations with commas. (Optional)

–instances *name1*,*name2*
The session management server instances which are to be administered. The instance names must be separated by a comma. The default value is `DSess`. (Optional)

-ssl_enable {yes|no}
Indicates whether SSL communication with the web server must be enabled. (Optional)

–sslkeyfile *path*
Specifies the fully qualified name of the SSL key file to use during communication with the session management server web service. Use this parameter only when the `-ssl_enable` parameter is set to `yes`. (Optional)

–sslkeyfile_label *label*
Specifies the SSL key file label of the certificate to be used. Use this parameter only when the -ssl_enable parameter is set to yes. (Optional)

–sslkeyfile_stash *path*
Specifies the fully qualified name of the stash file that contains the password for the SSL key file. Use this parameter only when the -ssl_enable parameter is set to yes. (Optional)

Availability

This utility is in one of the following default installation directories:

- On AIX, Linux, and Solaris operating systems:
  /opt/pdsms/bin
- On Windows operating systems:
  c:\Program Files\Tivoli\PDSMS\bin

To start the command line under Windows, use **pdsmsclicfg-cl.exe**. The **pdsmsclicfg** command starts the wizard.

When an installation directory other than the default is selected, this utility is in the /bin directory under the installation directory (for example, *installation_directory*/bin).

Return codes

0 The utility completed successfully.

non-zero
The utility failed. When a utility fails, a description of the error and an error status code in hexadecimal format is provided (for example, 0x15c3a00c). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

smsbackup

Gathers information to help IBM Software Support in problem determination.

Note: This utility is for use by support personnel.

Syntax

For local mode
smsbackup –local [–was_home *path*] [–list *list*] [–path *output*]

For remote mode
smsbackup [–was_home *path*] [–wsadmin_options *options*] [–list *list*] [–path *output*]

Description

The **smsbackup** gathers information to help IBM Software Support in problem determination. It has two modes:

Local mode
Gathers information from the local system only. Does not require an

operational WebSphere environment. To gather information about all members in the cluster, run the utility on each node in the cluster.

Remote mode
Gathers information about the entire environment. Requires an operational WebSphere environment.

The utility is provided on AIX, Linux, and Solaris operating systems as a shell script, **smsbackup.sh**. On Windows operating systems, it is provided as a batch script, **smsbackup.bat**.

When the utility runs in local mode, you must copy the following files to each member in the cluster, maintaining directory structure:

- /bin/smsbackup.sh
- /bin/smsbackup.bat
- /etc/smsbackup.lst
- /lib/smscfg.jar
- /nls/java/message.jar

Parameters

–local Indicates that the utility runs in local mode.

–list *list*
Specifies the .lst file that describes the information to gather. If not specified, the smsbackup.lst file in the *sms_installation_directory*/etc directory is used. (Optional)

–path *output*
Specifies the directory for the created JAR file. The JAR file contains the gathered information. (Optional)

–was_home *path*
Specifies the home directory of the WebSphere Application Server. This value must be set on the command line or in the WAS_HOME environment variable. (Optional)

–wsadmin_options *options*
Specifies options to pass directory to the **wsadmin** utility. Use this parameter to pass non-default binding information before the backup operation runs through the WebSphere cluster. Examples of non-default binding information include the user name and password. (Optional)

Availability

This utility is in one of the following default installation directories:

- On AIX, Linux, and Solaris operating systems:
  /opt/pdsms/bin
- On Windows operating systems:
  c:\Program Files\Tivoli\PDSMS\bin

When an installation directory other than the default is selected, this utility is in the /bin directory under the installation directory (for example, *installation_directory*/bin).

Return codes

0 The utility completed successfully.

non-zero

The utility failed. When a utility fails, a description of the error and an error status code in hexadecimal format is provided (for example, 0x15c3a00c). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

smscfg

Deploys and configures the session management server.

Syntax

smscfg –action {config|unconfig|deploy|undeploy|extract|upgrade|revert}

Configuration

smscfg –action config [–interactive {yes|no}] [–rsp_file *file_name*] [–record *file_name*] [–was_port *port*] [–was_enable_security {yes|no}] [–was_admin_id *administrator_id*] [–was_admin_pwd *password*] [–trust_store *file_name*] [–trust_store_pwd *password*] [–keyfile *file_name*] [–key_pwd *password*] [–instance *instance_name*] [–session_realm *realm*:*max_login*=*replica_set1_name*,*replica_set2_name*,...] [–session_realm_remove *realm_name*] [–enable_tcd {yes|no}] [–tcd *fully_qualified_directory_name*] [–enable_sam_integration {yes|no}] [–policysvr_host *host_name*] [–policysvr_port *port*] [–admin_id *administrator_id*] –admin_pwd[*password*] [–domain *domain*] [–authzsvr *host_name*:*port*:*rank*] [–cred_refresh_rule *rule*] [–enable_last_login {yes|no}][–enable_last_login_database {yes|no}] [–last_login_table *last_login_database_table_name*] [–last_login_max_entries *max_number_memory_entries*] [–last_login_jsp_file *file_name*] [–last_login_jsp *server_jsp_name*][–enable_database_session_storage {yes|no}][–enable_auditing {yes|no}][–auditing_properties *file_name*][–key_lifetime *key_lifetime*] [–client_idle_timeout *timeout*]

Configuration with response file

smscfg –action config –rspfile *file_name*

Configuration, interactive

smscfg –action config –interactive

Unconfiguration

smscfg –action unconfig [–interactive {yes|no}] [–rspfile *file_name*] [–record *file_name*] [–was_port *port*] [–was_enable_security {yes|no}] [–was_admin_id *administrator_id*] [–was_admin_pwd *password*] [–trust_store *file_name*] [–trust_store_pwd *password*] [–keyfile *file_name*] [–key_pwd *password*] [–instance *instance_name*] [–admin_id *administrator_id*] [–admin_pwd *password*] [–remove_last_login_db {yes|no}]

Unconfiguration, response file

smscfg –action unconfig –rspfile *file_name*

Unconfiguration, interactive

smscfg –action unconfig –interactive

Deployment

smscfg –action deploy [–interactive {yes|no}] [–rspfile *file_name*] [–record *file_name*] [–was_port *port*] [–was_enable_security {yes|no}] [–was_admin_id *administrator_id*] [–was_admin_pwd *password*]

```
[–trust_store file_name] [–trust_store_pwd password] [–keyfile
file_name] [–key_pwd password] [–instance instance_name]
[–enable_database_storage {yes|no}][–database_name
database_name][–virtual_host host_name] [–clustered {yes|no}] [–was_node
node_name] [–was_server server_name] [–was_cluster cluster_name]
```

Undeployment

```
smscfg –action undeploy [–interactive {yes|no}] [–rspfile file_name]
[–record file_name] [–was_port port] [–was_enable_security {yes|no}]
[–was_admin_id administrator_id] [–was_admin_pwd password]
[–trust_store file_name] [–trust_store_pwd password] [–keyfile
file_name] [–key_pwd password] [–instance instance_name]
```

Extract

```
smscfg –action extract [–interactive {yes|no}] [–rspfile file_name]
[–record file_name] [–was_port port] [–was_enable_security {yes|no}]
[–was_admin_id administrator_id] [–was_admin_pwd password]
[–trust_store file_name] [–trust_store_pwd password] [–keyfile
file_name] [–key_pwd password] [–instance instance_name]
```

Upgrade

```
smscfg –action upgrade [–interactive {yes|no}] [–rspfile file_name]
[–record file_name] [–was_port port] [–was_enable_security {yes|no}]
[–was_admin_id administrator_id] [–was_admin_pwd password]
[–trust_store file_name] [–trust_store_pwd password] [–keyfile
file_name] [–key_pwd password] [–instance instance_name]
```

Revert

```
smscfg –action revert [–interactive {yes|no}] [–rspfile file_name]
[–record file_name] [–was_port port] [–was_enable_security {yes|no}]
[–was_admin_id administrator_id] [–was_admin_pwd password]
[–trust_store file_name] [–trust_store_pwd password] [–keyfile
file_name] [–key_pwd password] [–instance instance_name]
```

Utility help

```
smscfg –help option
```

```
smscfg –usage
```

```
smscfg –?
```

Description

The **smscfg** utility deploys, configures, or unconfigures session management server instances. It can also be used to extract the session management server configuration, or to install and remove fix pack upgrades.

A log of the configuration progress is written to `msg_smscfg.log` log file. This log file is in:

- The `/var/pdsms/log` directory on AIX, Linux, and Solaris operating systems.
- The *installation_directory*`\log` directory on Windows operating systems.

This utility can be run in one of these ways:

- Interactively – the user is prompted to provide configuration information.
- Silently – the utility accepts input from a response file.

Parameters

–? Displays the syntax and an example for this utility. (Optional)

–action {**deploy**|**config**|**unconfig**|**undeploy**|**extract**}
Specifies the action that is one of the following values:

deploy
Deploys the session management server instance to a WebSphere Application Server.

undeploy
Removes a session management server instance from a WebSphere Application Server.

config Configures or reconfigures a deployed session management server instance.

unconfig
Unconfigures a session management server instance.

extract Extracts the configuration information from a session management server instance.

upgrade
Upgrades to a new session management server fix pack.

revert Reverts to the previous session management server fix pack.

–admin_id *administrator_id*
Specifies the Security Access Manager administration ID. The default value is `sec_master`. This parameter is required when `–enable_sam_integration` is set to `yes`. (Optional)

–admin_pwd *password*
Specifies the password for the Security Access Manager administrator. This parameter is required when you specify the `–admin_id` parameter. (Optional)

–auditing_properties *file_name*
Specifies the path to the properties file which contains the configuration of the auditing component. (Optional)

–authzsvr *host_name*:*port*:*rank*
Specifies the host name, port number, and rank of the Security Access Manager authorization server. This optional parameter can be specified multiple times.

A Security Access Manager authorization server is required to use either of these functionalities:

- Session refresh capabilities
- Certificates that are issued by the Security Access Manager policy server to authenticate session management clients

The default value is `localhost:7136:1`. (Optional)

–client_idle_timeout *timeout*
Specifies the client idle timeout in seconds after which a client is considered idle. A client is considered idle if it is not actively requesting updates from the session management server. (Optional)

–clustered {**yes**|**no**}
Whether the application is deployed to a WebSphere cluster. The default value is `no`. (Optional)

–cred_refresh_rule *rule*
: Specifies rules to preserve when a user credential is refreshed. The default credential refresh rule set is `preserve=tagvalue_*`. (Optional)

–database_name *database*
: Specifies the name of the WebSphere JDBC data source. The session management server uses this data source to access the database where the server stores its data. There is no default value. (Optional)

–domain *domain*
: Specifies the name of the Security Access Manager policy domain. This parameter is required when `–enable_sam_integration` is set to `yes`. The default value is `Default`. (Optional)

–enable_auditing {yes|no}
: Indicates whether auditing is required. The default value is `no`. (Optional)

–enable_database_storage {yes|no}
: Indicates whether database storage is required. The parameter is only meaningful in the context of WebSphere Application Server single server deployments. If the application is deployed to a cluster, this parameter is redundant. The default value is `no`. Setting this parameter to `no` sets the database configuration to the WebSphere default resource reference, normally `jdbc/DataSource`. (Optional)

–enable_database_session_storage {yes|no}
: Indicates whether storage of session data to a database is required. The default value is `no`. (Optional)

–enable_last_login {yes|no}
: Indicates whether last login information is stored. When set to `yes`, you must specify the following parameters or accept their default values:

- `–last_login_jsp_file`
- `–last_login_max_entries`
- `–last_login_table`

The default value is `no` (not to enable the recording of last login information). The `–enable_last_login` field is only required if you install into a stand-alone application server. When you install into a cluster this field is not required. (Optional)

–enable_last_login_database {yes|no}
: Indicates whether last login information is stored to a database. The default value is `no`. (Optional)

–enable_tam_integration {yes|no}
: Indicates whether to enable integration with Security Access Manager or to change enablement. When set to `yes`, you must specify the following parameters or accept their default values, where applicable:

- `–policysvr_host`
- `–policysvr_port`
- `–authzsvr`
- `–admin_id`
- `–admin_pwd`
- `–domain`

The default value is `no`. (Optional)

–enable_tcd {**yes**|**no**}
: Indicates whether Tivoli Common Directory logging is required. When set to `yes`, you must specify the `–tcd` parameter. The default value is `no`. (Optional)

–help [*options*]
: Lists the name of the utility parameter and a short description. If one or more options are specified, it lists each parameter and a short description. (Optional)

–instance *instance_name*
: Specifies the name of the instance to be administered. The default value is `DSess`. (Optional)

–interactive {**yes**|**no**}
: Indicates whether the configuration is interactive. The default value is `yes`. (Optional)

–key_lifetime *lifecycle*
: Specifies the lifetime in seconds of the key for the session management server. After the defined lifecycle completes, a new key is generated. If this value is set to zero, keys are not automatically generated. (Optional)

–key_pwd *password*
: Specifies the password to access the server-side certificates. This parameter is required when you specify the `–keyfile` parameter. (Optional)

–keyfile *file_name*
: Specifies the fully qualified name for the keystore when a secure connection is made to WebSphere Application Server. The keystore holds the server-side certificates. This parameter is required when you specify the `–was_admin_id` parameter. (Optional)

–last_login_jsp *server_jsp_name*
: The server-side path for the last login JSP file. *server_jsp_name* is an optional argument. (Optional)

–last_login_jsp_file *file_name*
: Specifies the fully qualified name of the last login JSP file to use for recording last login information. This parameter is required when the `–enable_last_login` parameter is set to `yes`. The default value is *installation_directory*`/etc/lastLogin.jsp`. (Optional)

  Note: Configuration of the `lastLogin.jsp` file can produce a long web browser URL, which can exceed the limits that are imposed by some proxy servers. Access the WebSphere ISC by using a direct connection to the Internet.

–last_login_max_entries *maximum_entries*
: Specifies the maximum number of entries to be stored in the memory cache for recording last login information. This parameter is required when the `–enable_last_login` parameter is set to `yes`. The default value is `0`. The `–last_login_max_entries` field is only required when you install into a stand-alone application server. When you install into a cluster, this field is not required. (Optional)

–last_login_table *table_name*
: Specifies the name of the database table to use for recording last login information. This parameter is required when the `–enable_last_login` parameter is set to `yes`. The default value is `AMSMSUSERINFOTABLE`. (Optional)

–operations
Lists each of the parameter names, one after another, without a description. (Optional)

–policysvr_host *host_name*
Specifies the host name of the Security Access Manager policy server. This parameter is required when `–enable_sam_integration` is set to `yes`. (Optional)

–policysvr_port *port*
Specifies the port of the Security Access Manager policy server. This parameter is required when you specify the `–host` parameter. (Optional)

–record *file_name*
Specifies the name of the response file to which configuration parameters are recorded. (Optional)

–remove_last_login_db **{yes|no}**
Indicates whether the last login database must be removed. The default value is `no`. (Optional)

–rspfile *response_file*
Specifies the fully qualified path and file name of the response file to use during silent configuration. A response file can be used for configuration. There is no default response file name. The response file contains stanzas and *key*=*value* pairs. For information about using response files, see the "Using response files" appendix in the *IBM Security Access Manager for Web: Command Reference*. (Optional)

–session_realm **[***realm***[:***max_logins***]=***replica_set1***,** *replica_set2***,...]**
Specifies a session realm to add to the configuration. If the session realm name or any of the replica set names contain spaces, the entire argument must be specified within quotation marks.

The `max_logins` parameter is used to specify the maximum number of concurrent login events that are permitted for the session realm. If the `max_logins` parameter is not supplied, there are an unlimited number of concurrent login events that are allowed for the session realm.

Replica set names must be separated by commas. (Optional)

–session_realm_remove *realm*=*set_name***[,...][;***realm*=*set_name***[,...]...]**
Specifies the name of a session realm to remove. If the session realm name contain spaces, the entire argument must be specified within quotation marks. (Optional)

–tcd *path_name*
Specifies the fully qualified directory for Tivoli Common Directory logging. This parameter is required when `–enable_tcd` is set to `yes`. If the Tivoli Common Directory is configured on the target system, this option is ignored. (Optional)

–trust_store *file_name*
Specifies the fully qualified name for the truststore when a secure connection is made to WebSphere Application Server. The truststore holds the client-side certificates. This parameter is required when you specify the `–was_admin_id` parameter. (Optional)

–trust_store_pwd *password*
Specifies the password to access the client-side certificates. This parameter is required when you specify the `–trust_store` parameter. (Optional)

–usage Displays the syntax and an example for this utility. (Optional)

–virtual_host *host_name*
Specifies the name of the WebSphere virtual host to which to deploy the session management server application. If not specified, the application is deployed on the default virtual host. (Optional)

–was_admin_id *administrator_id*
Specifies the name of the administrator to use when a secure connection is made to WebSphere Application Server. In interactive mode, this parameter is optional unless you are making a secure connection. When you use this parameter, you must specify the `–was_admin_pwd` parameter. When not making a secure connection, this parameter is optional. (Optional)

–was_admin_pwd *password*
Specifies the password to use when a secure connection is made to WebSphere Application Server. The administrator can use this password. (Optional)

–was_cluster *cluster_name*
Specifies the name of the WebSphere cluster to which to deploy the session management server application. This parameter is mutually exclusive with the `–was_server` parameter. (Optional)

When you are using WebSphere Network Deployment and `–was_cluster` is specified, and there is only one cluster, the application is deployed to that cluster.

The application is deployed to that server when you are using WebSphere Network Deployment and:

- `–was_cluster` is specified.
- There is no cluster.
- There is only one server.

–was_enable_security {**yes**|**no**}
Indicates whether the communication with the WebSphere server uses a secure connection. When set to `yes`, you must specify the following parameters:

- `–was_admin_id`
- `–was_admin_pwd`
- `–trust_store`
- `–trust_store_pwd`
- `–keyfile`
- `–key_pwd`

The default value is `no`. (Optional)

–was_node *node_name*
Specifies the name of the WebSphere node. (Optional)

–was_port *port*
Specifies the SOAP port to use on the WebSphere server. This parameter is always required unless the `–interactive` parameter is set to `yes`.

–was_server *server_name*
Specifies the name of the WebSphere server to which to deploy the session management server application. This parameter is mutually exclusive with the `–was_cluster` parameter. (Optional) The application is deployed to the server to which this configuration utility is connected when:

- WebSphere Application Server is in a single server deployment, and

- `–was_server` is not specified.

Availability

This utility is in one of the following default installation directories:

- On AIX, Linux, and Solaris operating systems:

  `/opt/pdsms/bin`
- On Windows operating systems:

  `c:\Program Files\Tivoli\PDSMS\bin`

When an installation directory other than the default is selected, this utility is in the `/bin` directory under the installation directory (for example, *`installation_directory`*`/bin`).

Return codes

0 The utility completed successfully.

non-zero
The utility failed. When a utility fails, a description of the error and an error status code in hexadecimal format is provided (for example, `0x15c3a00c`). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

smsservicelevel

Lists the current service level of the session management server files on the local system.

Note: This utility is for use by support personnel.

Syntax

`smsservicelevel` [*`directory`* [*`directory`*] ...] [*`file`* [*`file`*] ...]

Description

The **`smsservicelevel`** utility recursively scans the specified directory. The utility returns to the standard output device the name and service level for session management server files. The files match Security Access Manager conventions.

The utility is provided on AIX, Linux, and Solaris operating systems as a shell script, **`smsservicelevel.sh`**. On Windows operating systems, it is provided as a batch script, **`smsservicelevel.bat`**.

Parameters

`directory`
Specifies the directories that the utility searches for service level information. (Optional)

`files` Specifies particular files that the utility searches. (Optional)

Availability

This utility is in one of the following default installation directories:

- On AIX, Linux, and Solaris operating systems:

  `/opt/pdsms/bin`
- On Windows operating systems:

  `c:\Program Files\Tivoli\PDSMS\bin`

When an installation directory other than the default is selected, this utility is in the `/bin` directory under the installation directory (for example, *`installation_directory`*`/bin`).

Return codes

0 The utility completed successfully.

non-zero

The utility failed. When a utility fails, a description of the error and an error status code in hexadecimal format is provided (for example, `0x15c3a00c`). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

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

Printed in USA

SC23-6509-02