WebSphere® IBM WebSphere Presence Server

Version 7.0

IBM WebSphere Presence Server

Second Edition (April 2010)

This edition applies to IBM WebSphere Presence Server, Version 7.0, and to all subsequent releases and modifications until otherwise indicated in new editions.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
Department 6R4A
P.O. Box 12195
Research Triangle Park, North Carolina
27709-2195

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Chapter 1. Introduction to IBM WebSphere Presence Server Component 1

Overview 1
What's new in Presence Server version 7.0 1
Features of the Presence Server product 3
Mapping and normalization 3
Content indirection 4
PUBLISH and SUBSCRIBE flows 5
SIP external sources 6
The Presence Server entry point 7
Presence authorization rules 10
Watcher information 12
IBM WebSphere Presence Server REST interface 13
Notification throttling 13
Presence data storage 14
The Presence data model 15
Interactions with other IBM WebSphere software for Telecom 17
Conformance with industry standards 17

Chapter 2. Planning for IBM WebSphere Presence Server Component 19

Hardware and software requirements. 19
Hardware requirements 19
Software requirements. 20
Planning to install IBM WebSphere Presence Server Component 21
Evaluating your hardware environment 22
Integrating with supported network elements . . . 23
Planning for the databases 24
Planning for high availability 25
Planning security for Presence Server. 25
General considerations for setting up security for Presence Server 25
Planning authentication security using the Trust Association Interceptor 28
Planning to migrate from a previous release of IBM WebSphere Presence Server Component 30

Chapter 3. Installing IBM WebSphere Presence Server Component 33

Migrating from a previous release of IBM WebSphere Presence Server Component 33
Preparing the environment 34
Verifying that the required software is installed 34
Creating the cluster. 35
Defining the proxy server 37
Preparing the Trust Association Interceptor for use 38
Configuring WebSphere security for the Trust Association Interceptor 38
Preparing the installation files for the Trust Association Interceptor 38
Preparing the databases 39
Creating the Presence Server database for DB2 . 39
Creating the content indirection database for DB2 41
Creating the usage records database for DB2 . . 43
Creating the Service Integration Bus (SIBus) database for DB2 45
Creating Presence Server databases for Oracle . . 46
Installing the Presence Server product 46
Installing Presence Server using the interactive installer. 46
Installing Presence Server silently 49
Setting replication for the SIP container 55
Installing updates for the Presence Server product. 56
Adding a server to an existing Presence Server node 58
Uninstalling IBM WebSphere Presence Server Component 62
Uninstalling Presence Server using the interactive uninstaller. 62
Uninstalling Presence Server silently 62
Dropping the databases 64

Chapter 4. Configuring IBM WebSphere Presence Server Component 73

Configuring for integration with external sources. . 73
Configuring for SIP external sources 73
Configuring Presence Server to interact with S-CSCF sources 76
Configuring for integration with IBM XDMS . . . 79
Configuring authorization and authentication . . . 82
Configuring the way in which PUBLISH and SUBSCRIBE requests are handled 88
Configuring partial publish and partial notify . . 88
Configuring expiration times for PUBLISH and SUBSCRIBE requests 89
Adding headers to outgoing NOTIFY requests. . 91
Configuring content indirection. 93
Configuring routing using the Presence Server entry point 95
Configuring watcher information 97
Configuring performance-related settings 99
Configuring usage records 103
Configuring error notification 104
Configuring the REST interface to Presence Server 106
The XML configuration file 107

Chapter 5. Administering the IBM WebSphere Presence Server Component 113

Restarting Presence Server 113
Stopping and starting the server 113
Stopping a cluster 114
Stopping a server (console) 114
Stopping a server (command line) 115
Stopping the node agent (console) 115
Stopping the node agent (command line) . . . 116

Stopping the deployment manager (console) . . 116
Stopping the deployment manager (command line) 116
Starting the deployment manager. 117
Starting the node agents 117
Starting a cluster 117
Starting a server (console) 118
Starting a server (command line) 118
Monitoring system performance using WebSphere PMI 119
Enabling performance monitoring 119
Disabling performance monitoring 120
Performance metrics 120
Modifying logging 124
Using IBM Tivoli License Manager 126

Chapter 6. Troubleshooting IBM WebSphere Presence Server Component 129
Using ISA 4.0 add-ons to communicate with IBM Support 129
Troubleshooting the database script 129
Monitoring log messages 130
Viewing and modifying logs 131
Enabling trace 132
Selecting trace loggers 134
Messages 135
Message key. 135

Chapter 7. Developing applications and using product features 137
Extending Presence Server authorization 137
General information about authorization . . . 137
Developing an authorization application . . . 138
Adding the API JAR file to the class path . . . 140
Using the REST API sample 140
Description of the REST API sample. 141
Installing and configuring the REST API sample 141
Running the REST API sample 141
Using the SIP sample. 142
Description of the SIP sample 142
Installing and configuring the SIP sample . . . 143
Running the SIP sample 143
Starting a custom REST project 144
Migrating your applications from a previous release of IBM WebSphere Presence Server Component 144
Accessing data from SIP external sources 145

Chapter 8. Reference information . . . 147
Changes to this edition 147
Documentation conventions 147
Directory conventions 147
Glossary 148

Notices 159
Trademarks 160

Chapter 1. Introduction to IBM WebSphere Presence Server Component

The IBM® WebSphere® Presence Server Component enhances applications with consolidated, real-time information regarding user availability, capability, and willingness to communicate.

Presence Server can collect and distribute presence information to enable awareness between users in the enterprise. Applications can use it to publish presence information.

Applications can also subscribe to presence information. When an application subscribes to presence information, Presence Server sends notifications, containing the presence information, to the application. Presence Server manages the presence information and maintains it in a database.

Presence Server is deployed as an enterprise application in a WebSphere Application Server environment.

Overview

IBM WebSphere Presence Server Component supports SIP protocol to provide awareness of presence. The application provides a set of SIP servlets and a representational state transfer (REST) interface for handling PUBLISH and SUBSCRIBE requests using either SIP or HTTP.

For SIP Presence Server handles application client requests with a SIP proxy, which passes them to the SIP container. The SIP container analyzes them and sends them to WebSphere Presence Server through SIP servlets. For HTTP requests are handled by an HTTP proxy which passes them to an HTTP container. The HTTP container analyzes them and send them to WebSphere Presence Server through the HTTP servlets.

Presence Server collects, manages, and distributes standard presence information Presence Information Data Format (PIDF). The information contains one or more sets of data. Each data set consists of a status indicator, an optional communication address, and other optional presence information. Users can subscribe to presence information for other users, receive information about other users, or receive notification when that information changes.

Presence Server obtains presence information from the client applications and external presence providers, including external registrars.

What's new in Presence Server version 7.0

IBM WebSphere Presence Server version 7.0 includes new functionality.

Notification throttling

Presence Server implements the notification throttling mechanism as specified in IETF draft-niemi-sipping-event-throttle-06. This feature can help improve performance by limiting the rate of SIP event notifications for one or more event packages.

Presence Server REST interface

Presence Server exposes a representational state transfer (REST) interface for presence operations over HTTP, allowing for fetch and publish presence operations using a Web client.

The REST interface permits Web clients to perform HTTP GET, PUT, POST, and DELETE requests.

With this feature it is possible for SIP clients and clients that use the REST interface to operate together, when all of the clients are connected to the same server. In particular, presence authorization rules are respected and applied both for fetch operations by HTTP clients and for information published by HTTP clients.

Public identity mapping

Using public identity mapping, Presence Server can tie together presence information from disparate sources that relate to the same person or presentity but use different, non-correlated identifiers. With identity mapping, presence information from different IDs for the same user are presented as a single aggregated presence document. When a subscriber subscribes to any one of these identities, the subscriber receives the aggregated document.

Support for local lists

To facilitate integration with the IBM XDMS product, Presence Server supports the use of `<rls-services>` documents–in particular, local lists in RLS services documents, as specified in RFC 4826.

Subscriber based partitioning and RLS (Presence Server entry point)

The Presence Server entry point provides ways for handling incoming PUBLISH and SUBSCRIBES more efficiently.

You can deploy several Presence Server clusters, each of which handles SIP requests for a specified subset of users. This facilitates scaling so that you can support a large number of users. Presence Server contains a routing component that routes each request to the correct cluster.

You can also designate a separate resource list server (RLS) component to handle all subscriptions on Presence lists, as described in the standard OMA-TS-Presence_SIMPLE-V1_1-20080627-A, chapter 5.5. This use of the RLS is designed to reduce network traffic and improve performance by reducing the number of subscriptions that have to be established between the RLS cluster and the Presence Server clusters.

Presence data model

Presence Server conforms to the Presence data model as specified in RFC 4479. The data model stipulates that each presentity should have only one document, and–if possible–each document should have only one `<person>` element. It also stipulates that the watcher, not the presence composer, should resolve ambiguities when merging information from multiple sources.

Content indirection

Presence Server supports content indirection as described in RFC 4483, allowing a SIP message to contain an indirect reference to the desired content. The receiving party then uses this indirect reference to retrieve the content by means of a non-SIP transfer channel such as HTTP. An alternative transport mechanism is provided for SIP body parts.

Interactive installation

A simplified, interactive process is provided for installing the Presence Server product. The installation package supports silent installation using a response file, and it can also be used for uninstalling and reinstalling the product. The installation package can be used for all of the supported hardware topologies.

Features of the Presence Server product

To enable awareness between users in the enterprise, Presence Server offers a number of different features.

Mapping and normalization

Using public identity mapping, Presence Server can tie together presence information from disparate sources that relate to the same person or presentity but use different, non-correlated identifiers. With identity mapping, presence information from different IDs for the same user are presented as a single aggregated presence document. When a subscriber subscribes to any one of these identities, the subscriber receives the aggregated document.

Presence Server begins by preloading the contents of white and black lists into memory. If public identity mapping is enabled for the white or black lists, Presence Server maps each identity to its fundamental identity and then manages it in memory. For incoming PUBLISH and SUBSCRIBE requests, the publisher/subscriber identity is mapped and the fundamental identity is compared to the identities in the white or black list. When the lists are updated (by default once an hour) the identities are mapped again.

PUBLISH operations

For PUBLISH operations, Presence Server verifies that the publisher is publishing his or her own presence document (unless the publisher in the white list). The identity comparison is based on the UIDNormalizer method for comparison identities without referring to their fundamental identity. The comparison is applied for the sender identity as taken from the WebSphere Application Server principal (from the p-asserted-identity header) for the ″To″ header and for the entity attribute as specified in the presence document.

The published document is stored in the database based on the presentity's fundamental identity. Presence information published from different identities is aggregated into a single document according to their fundamental identity. In this way, a subscriber who is subscribing to any one of the user's public identities will receive the full information published for the presentity. As a result, it is possible to perform a modify/unpublish operation with a continued eTag from different public identities that map to the same fundamental ID. For example: user A publishes document X, user B can modify document X, provided A and B map to the same fundamental identity.

Note: In some cases Presence Server needs to compare contact addresses, mainly in merges of PIDF documents. For this purpose Presence Server uses case-sensitive string comparison only.

Presence Server manages all incoming subscriptions based on the fundamental identity of the user the subscription is on (as taken from the To header). Upon re-subscribing, the presentity is mapped again to its fundamental identity, and if the mapping has changed Presence Server handles this as follows: a full notification is sent with the presence information for the new presentity, and the winfo notifications are also generated if necessary. For URI-list subscriptions or group subscriptions, all members of the groups are also mapped to their fundamental identities in the same manner.

SUBSCRIBE operations

For SUBSCRIBE operations, if presence rules are disabled and the subscriber is not in the white or black lists, Presence Server verifies that the subscriber is allowed by comparing the sender presentity from the WebSphere Application Serverprincipal (as taken from the p-asserted-identity header) with the presentity in the To header, without regard to fundamental identity, and allows the subscription.

When presence rules are enabled, by default Presence Server does not use identity mapping for testing identity conditions. Presence Server tests identity conditions matching using original subscriber identity by applying the UIDNormalizer method for comparison without regards to the fundamental identity. The domain comparison is done using UIDNormalizer as well. The domain of the original subscriber public identity is used for testing the domain condition. A configurable option is provided to apply identity mapping when testing identity conditions. With this option, Presence Server considers fundamental subscriber identity when using UIDNormalizer for testing identity conditions. The domain of the subscriber fundamental identity is used for testing the domain condition in this case.

For a winfo subscription, if the subscriber is not in the white or black list, Presence Server verifies that the subscriber is allowed by comparing the sender presentity from the WebSphere Application Server principal (as taken from the p-asserted-identity header) with the presentity in the To header, without regard to fundamental identity. A watcher that subscribes on any of its public identities receives the document with aggregated watchers for all presentities relating to the same fundamental identity.

The Telecom Application Enablement Feature provides samples. For more information, refer to the *IBM WebSphere Telecom Application Enablement Feature information center*.

Content indirection

Presence Server supports the content indirection standard, allowing a SIP message to contain an indirect reference to the desired content. The receiving party then uses this indirect reference to retrieve the content by means of a non-SIP transfer channel such as HTTP. An alternative transport mechanism is provided for SIP body parts.

Content indirection support is defined by IETF RFC 4483 and implemented by the Presence Server product as follows: the subscriber indicates in the SUBSCRIBE request that content indirection should be supported. The Accept header includes

the content indirection content type, for example: `Accept: message/external-body`. As a result, Presence Server marks the subscription to indicate that it supports content indirection.

Just before each SIP Notify message is sent to the subscriber, the size of the message body is checked. If the message body is larger than a configured size and the client supports content indirection, the following things take place:

- Presence Server creates a unique ID representing the current Notify body.
- The message body is saved in a database table with the Notify ID and the identity of the subscriber.
- The Notify message is altered so that it contains a content indirection reference to the saved Notify body. The reference is an HTTP URL to the content indirection Servlet with the content ID. The subscriber that receives the content indirection notification can use it to fetch the Notify body using HTTP GET request.

Note that this mechanism will always generate a unique notification ID for each outgoing notification, even if the notification content has not changed.

Content indirection servlet

The content indirection servlet handles HTTP GET requests for the indirect content. It extracts the content ID from the request and uses it to retrieve the document from the content indirection database. Assuming that the document is found and authorization tests are successful, the response with the full body is sent.

The authorization tests are based on the sender's identity, which is extracted from the HTTP request and translated to a WebSphere Application Server principal. The WebSphere Application Server principal is compared to the subscriber identity as stored in the database. If the identities are different, the servlet sends a 403 ″Forbidden″ response. Unauthenticated users are supported for Subscribe requests for Presence Server and for HTTP GET requests. Assuming that the same identity was used for both operations (Subscribe and the HTTP GET), the response with the document is sent back to the sender.

Content indirection failover support

For each incoming subscription, Presence Server stores the accepted types as indicated in the Subscribe request on the SIP application session. In case of failover, the SIP application session is migrated to another server in the cluster, and this information is used to determine whether content indirection support is required for the subscription.

For HTTP GET requests, the content indirection HTTP URL (as specified in the notify request) routes the request by the HTTP proxy. The HTTP proxy address is configurable. When one of the Presence Server instances in the cluster fails, the proxy directs the HTTP request to an active Presence Server instance. Because the indirected information is stored in the database, all servers can handle the HTTP requests.

PUBLISH and SUBSCRIBE flows

IBM WebSphere Presence Server Component supports PUBLISH and SUBSCRIBE requests. To reduce network traffic, you can configure Presence Server to support partial PUBLISH flows and partial notifications resulting from SUBSCRIBE requests.

PUBLISH is a client-generated request to create, modify, and remove an event state associated with the address of record (AOR). The event state is carried in the body of the PUBLISH request and maintained by Presence Server as a presence document using the following process.

1. Presence Server receives a PUBLISH request and stores it in the database.
2. Presence Server generates a full presence document and aggregates new data into the presence document.
3. The generated presence document is sent to watchers.

SUBSCRIBE is a client-generated request for a current state and state updates from a remote node. Presence Server uses the following process for SUBSCRIBE requests:

1. It receives a SUBSCRIBE request and stores the request in memory.
2. It notifies the subscriber and sends the latest presence document from the database.
3. When a subscribed presentity changes the presence information, it generates notifications to all subscribers.

You can configure Presence Server for partial publications and notifications, which can improve system throughput by reducing the amount of data that is transported over the network. A partial publication or notification sends only those parts of presence documents that have changed, rather than sending the entire documents.

Presence Server supports partial publications and notifications as defined by the following IETF draft standards:

draft-ietf-simple-partial-pidf-format-08

draft-ietf-simple-partial-notify-08

draft-ietf-simple-partial-publish-06

draft-ietf-simple-xml-patch-ops-02

SIP external sources

IBM WebSphere Presence Server Component interacts with SIP external sources to provide enriched presence information for users.

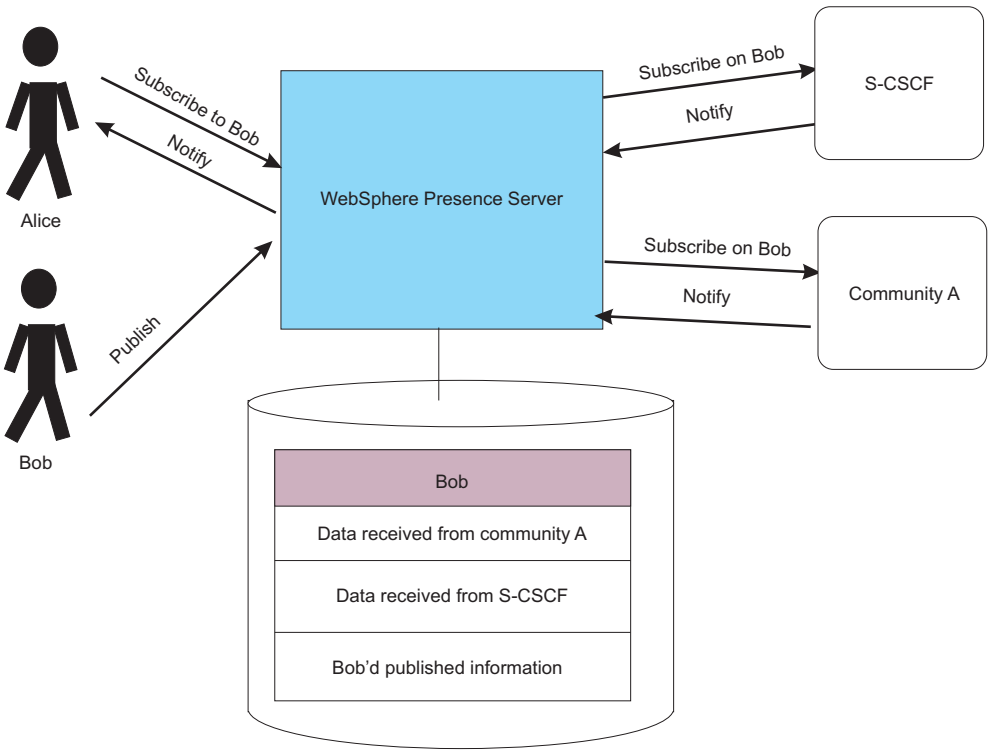
Presence Server interacts with SIP external sources such as S-CSCF and external communities. By receiving data from these external sources, Presence Server offers access to a richer set of presence information. An example of such information would be data about a given user in other communities besides the community represented by this instance of Presence Server.

In one common scenario, Presence Server collects presence information from an external registrar, S-CSCF, and aggregates the information with the published presence data. As a result, you have access to more information about presentities and their availability. This is especially useful when you are monitoring simple client applications that register with the IMS network but do not publish presence documents.

Aggregating presence information from SIP external sources also enables you to collect information about the presence of an individual user in multiple external communities and include this information in the presence document for this user in the local community.

When you use SIP external sources to interact with an external community, individual subscribers receive data according to presence authorization rules. The authorization rules are not enforced, however, for data that Presence Server receives from the external community. This happens because subscriptions to the external sources are based on a "super user" identity rather than on the identity of the original subscriber. However, the authorization is performed locally on all the presence information for the user (including information from external sources and external communities)

The following illustration shows the way in which Presence Server takes presence information from two external sources, in this case S-CSCF and an external community, and aggregates the information into a single presence document.



You can control the ways in which Presence Server interacts with SIP external sources. For details, see the topic *Configuring for integration with external sources*.

The Presence Server entry point

The Presence Server entry point provides ways for handling incoming PUBLISH and SUBSCRIBE requests more efficiently. You can designate a separate resource list server (RLS) component to handle all subscriptions on Presence lists requests, and you can set up a routing scheme in which different Presence Server clusters handle requests from different sets of users.

Collectively, this set of services is referred to as the Presence Server entry point.

Using the `SystemConfiguration.xml` file, you can enable a routing service for distributing requests among different Presence Server clusters. At the same time, you can also configure Presence Server to run in RLS mode. RLS mode is required if the routing service is enabled. For details, refer to the topic *Configuring routing using the Presence Server entry point*.

The following diagram shows the Presence Server and its relationship to other network elements in a typical configuration. In the diagram, an RLS cluster–representing the Presence Server entry point–receives traffic after it has passed through a DMZ or firewall and then directs the traffic to clusters in which Presence Server and IBM XDMS are running.

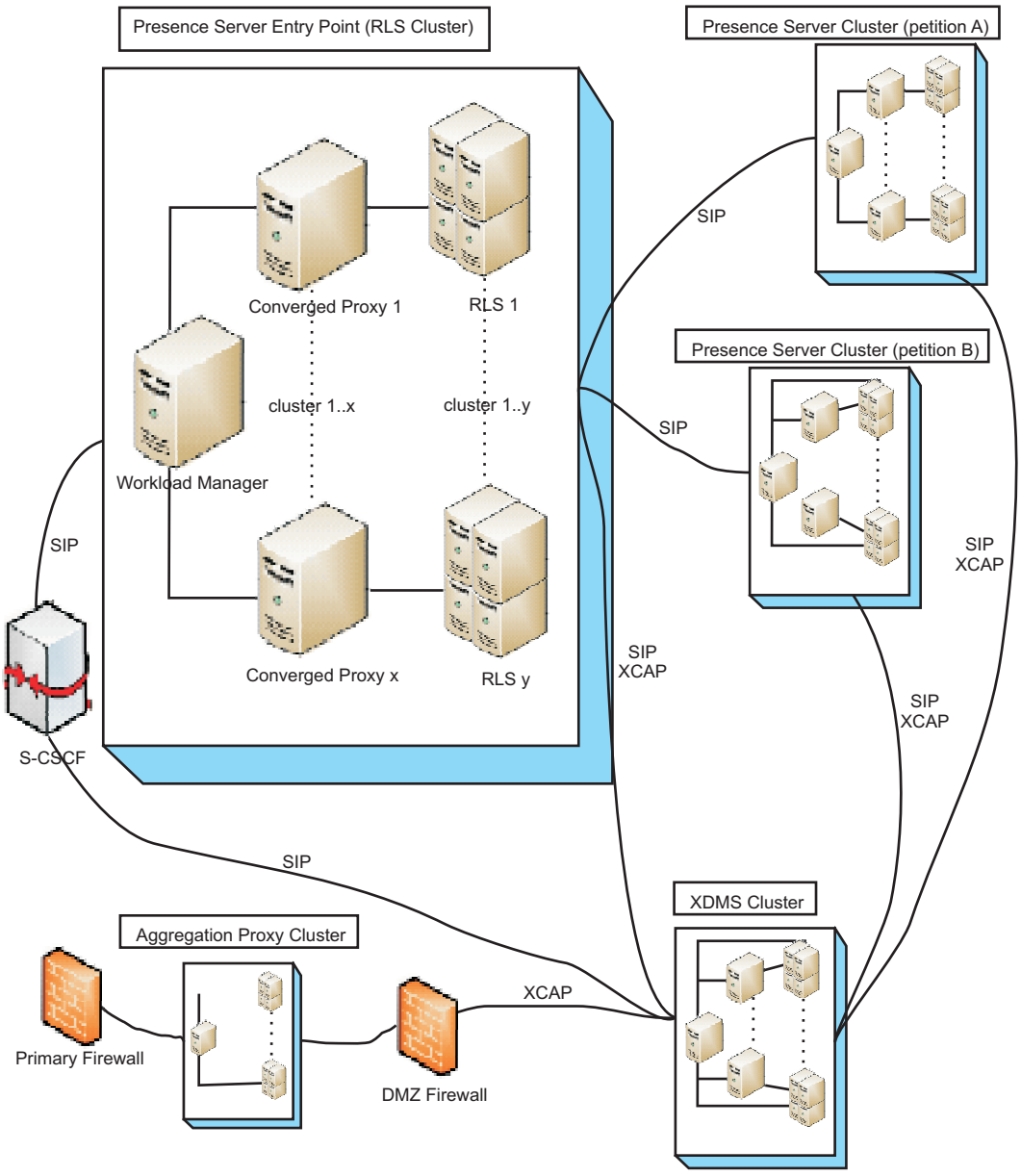


Figure 1. Presence Server entry point

Using routing to distribute SIP requests among clusters

You can deploy a number of different Presence Server clusters and have each one handle incoming SIP requests for a particular subset of users. This can facilitate workload balancing, especially as your network serves increasing numbers of users.

A routing component within the Presence Server application provides a default routing scheme by hashing user IDs. There is also a customizable interface (SPI)

that you can use if you would like to develop alternative routing algorithms. You use this interface by mapping each user's fundamental ID to the SIP address of one of your Presence Server instances.

The routing process is applied the first SUBSCRIBE request and to all incoming PUBLISH requests for a single resource. Note that the first SUBSCRIBE request passes through the RLS and is then routed to the Presence Server instance. Subsequent SUBSCRIBE requests (resubscribe and unsubscribe) go directly to the Presence Server instance. All PUBLISH requests pass through the RLS and are then routed to the Presence Server instance.

The routing process works as follows: Presence Server extracts the ″To″ URI from the request header to and uses it to calculate the fundamental ID of the user who is the object of the PUBLISH or SUBSCRIBE. The fundamental ID is then passed to the routing component, which determines the SIP address of the Presence Server cluster to which the request should be routed.

The routing component and the routing SPI do not preclude any external routing element that you might have defined between the network entry point application and your collection of Presence Server clusters.

Note that the routing options apply only to the following:

- Incoming PUBLISH and SUBSCRIBE requests on single resource, not on a list of resources
- SIP requests, not HTTP requests

Deploying a separate resource list server

You can deploy a separate resource list server (RLS) component to handle all subscriptions on Presence lists, as described in the standard OMA-TS-Presence_SIMPLE-V1_1-20080627-A, chapter 5.5.

When the RLS component receives a subscription on a resource list, it interacts with Shared list XDMS to receive the list members. For each list member, the RLS creates a back-end subscription to the network entry point–for example, the S-CSCF. The back-end subscription is routed to the correct network and then to the correct Presence Server by means of the routing component within the Presence Server application.

The back-end subscription is established on behalf of the RLS, and as such it will remain active as long as there is at least one incoming subscription that includes the list member. When notifications are received from Presence Server, the presence information is stored in the RLS cluster database and is used to establish future subscriptions that include this member. When the presence information for all list members is received, the RLS applies presence rules filtering, combines the RLMI document, and send a notification to the original subscriber.

Note: Resource List Meta-Information (RLMI), as defined in IETF RFC 4662, is used as the document format.

This use of the RLS is designed to reduce network traffic and improve performance by reducing the number of subscriptions that have to be established between the RLS cluster and the Presence Server clusters. So that Presence Server can continue to generate full watcher information for every user, the original

subscriber information is propagated by means of a PUBLISH request. The PUBLISH request is a proprietary message sent from the RLS to Presence Server.

This RLS component is deployed together with the routing component as a single application, referred to as the Presence Server entry point. The S-CSCF server forwards all presence related SIP requests to the Presence Server entry point, which performs the initial processing for each request.

Presence Server entry point does the initial processing of every incoming request. If the request is a PUBLISH or a SUBSCRIBE on a single resource, the request is routed to the correct Presence Server cluster. If the request is a SUBSCRIBE on a list of resources, the RLS component handles it.

The Presence Server entry point can handle HTTP requests as well. For each incoming HTTP request, Presence Server creates a matching SIP request and sends it to the S-CSCF. Alternatively, you can set up a cluster of Presence Server instances that handle all HTTP requests. The outgoing SIP messages are routed to the S-CSCF. When you use this approach, you reduce the burden on the RLS cluster and provide a separate mechanism for handling HTTP requests.

Presence authorization rules

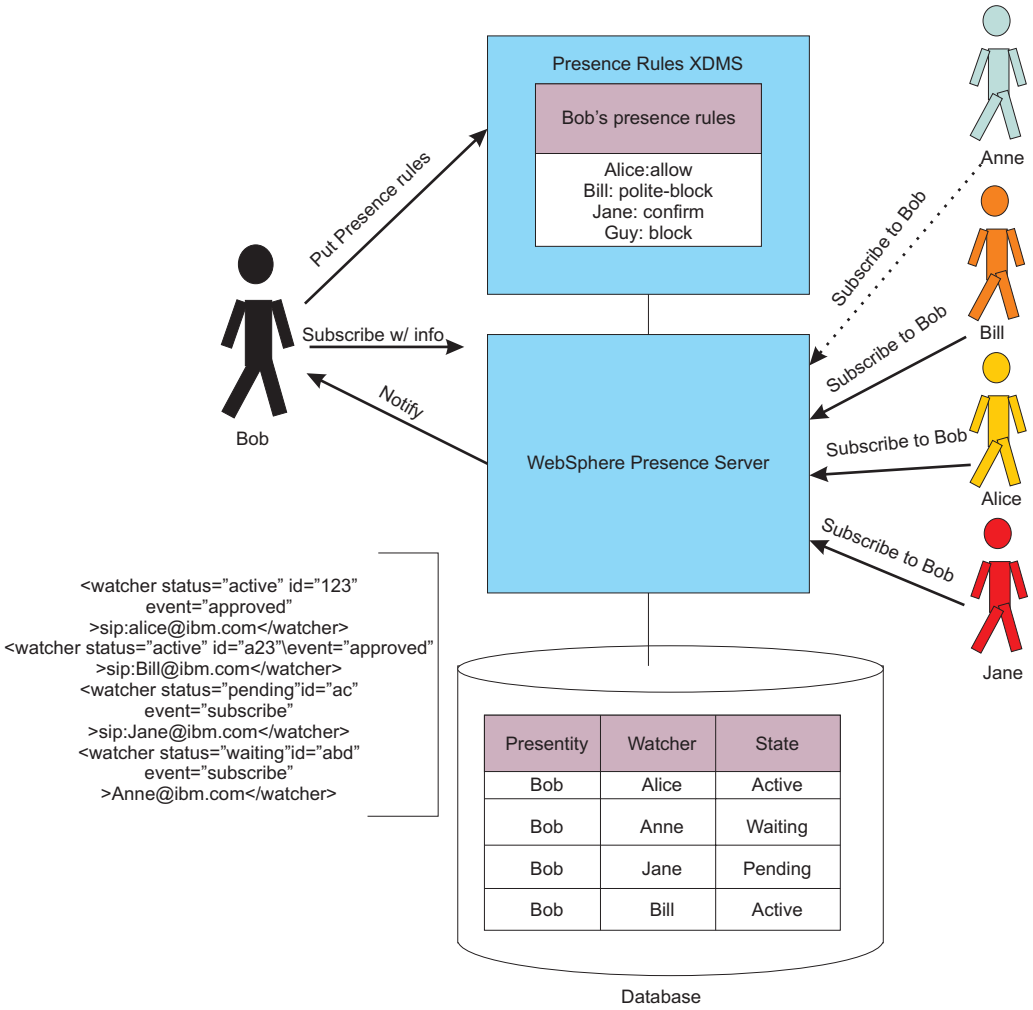
Presence authorization rules offer a highly flexible way for presentities to control who can access their presence information and what parts of the information are exposed.

Overview

Because presence information is sensitive, it is often a good practice to require authorization from a presentity (user) before the user's presence information can be sent to other subscribers. Presence authorization rules, using the format defined by IETF RFC 4745 and RFC 5025, give presentities a great deal of flexibility in controlling who has access to their presence information and what parts of the presence information are exposed.

Using authorization rules, Presence Server can evaluate every incoming SIP request and determine whether to allow the request. The request can either be allowed, polite-blocked, blocked, in a pending state awaiting confirmation that the sender is authorized, or in an undefined state. In each case an appropriate notification is returned to the sender.

The following illustration shows how one presentity (Bob) can use presence authorization rules determine the degree to which others can access his presence information. As he monitors watcher information, Bob can adjust his authorization rules as needed to extend to new watchers.



Samples for presence authorization rules, including conditions, actions, and transformations, are found in sections 6 and 7 of the IETF standard, *Presence Authorization Rules* (RFC 5025).

Configuring Presence Server to use authorization rules

To configure Presence Server so that it uses authorization rules, ensure that WebSphere Application Server Network Deployment application security is enabled. Then edit the `SystemConfiguration.xml` file and set `enable="true"` on the `presenceRules` tag. See the topic *Configuring authorization* for more details.

Note that, even when presence authorization rules are enabled, they can be overridden by white-list and black-list definitions. When white-list and black-list definitions are enabled, requests from subscribers on the white-list are always allowed, and requests from subscribers on the black-list are always blocked–regardless of what is specified in the authorization rules.

Authorization policies

As an alternative to authorization rules, you can develop custom authorization policies using the APIs provided as part of Presence Server. See the topic *Extending Presence Server authorization*.

For details about how Presence Server handles authentication and authorization for SIP requests, see the topic *Authentication Security*.

Watcher information

To facilitate presence authorization, IBM WebSphere Presence Server Component supports watcher information.

Support for watcher information in Presence Server

Presence Server supports watcher information as defined by IETF RFC 3857 and RFC 3858.

A watcher is defined as an entity that requests information about a resource, or user. These requests are made by means of subscriptions. The watcher information for any given resource is a list of all the subscriptions' status to that resource.

Watcher information consists of the current state for all subscriptions to a particular resource. Monitoring watcher information helps you control access to a resource because it provides the raw data you need to create and deploy presence authorization rules.

Examples of subscription states include active, waiting, pending, and terminated. Subscription state is dynamic, changing as users request new subscriptions, as old subscriptions expire, and as subscriptions are approved or rejected by the resource's owners. As a result, it often useful to subscribe to the watcher information for a particular resource.

Watcher information exists in an XML document that contains lists of watchers for a resource/event package pair. For every watcher in the list, the following attributes are included:

- The watcher's URI
- The watcher's display name (Optional)
- A unique identifier (ID) for the watcher
- Subscription status, for example active or pending
- The event that caused the transition to the current status
- Elapsed time since the SUBSCRIBE request that initiated the current subscription (Optional)

Interaction with other Presence Server features

The combination of watcher information and presence authorization rules enables the use of reactive authorization, where authorization occurs through direct user intervention. A user can subscribe to the watcher information for his or her presentity and thus find out when a new watcher is added who is not covered by the existing authorization rules. The user may then add a new authorization rule for the new watcher.

Presence Server also has configuration options to control what information is available to watchers, including white and black lists. Refer to the Configuring section of this information center for more information.

IBM WebSphere Presence Server REST interface

Presence Server exposes a representational state transfer (REST) interface for presence operations over HTTP, allowing for fetch and publish presence operations using a Web client.

The REST interface permits Web clients to perform HTTP GET, PUT, POST, and DELETE requests.

Using the REST interface it is possible for SIP clients and clients that use the REST interface to operate together, when all of the clients are connected to the same server. In particular, presence authorization rules are respected and applied both for fetch operations by HTTP clients and for information published by HTTP clients.

The Telecom Application Enablement Feature contains sample Web client code. For more information about the Web client sample and developing Web clients for use with the REST interface, refer to the *IBM WebSphere Telecom Application Enablement Feature information center*.

Fetch interface component

To fetch presence information about a resource, the client sends an HTTP GET request. The HTTP address consists of the server address, the presence prefix, and the presentity URI, for example, the email address.

An example of an address is `http://ps.example.com/presence/user@example.com`. The presence prefix is configurable, and can be changed by the administrator by changing the context root of the Presence Servlet in the `application.xml` file.

The HTTP response to a GET request adheres to the following characteristics:

- When successful the body of the response:

  returns as a pidf document in a case of get request on single user

  returns as a multipart related rlmi document if a list is requested
- When unsuccessful the body of the response:

  returns as empty in a case of get request on single user, if the user information is not found.

  returns the response code `404 document not found` if a requested list is not found.

Publish interface component

Presence Server allows Web clients to create, modify, and delete their presence information using HTTP POST, PUT and DELETE operations. The HTTP address for these operations is the same as in the fetch operation, that is, the presence prefix and the presentity URI, for example, `http://ps.example.com/presence/user@example.com`

Notification throttling

Presence Server version 7.0 implements the notification throttling mechanism as a means of improving performance.

Notification throttling is defined in IETF draft-niemi-sipping-event-throttle-06. It can help improve performance by limiting the rate of SIP event notifications for one or more event packages.

Presence Server provides a default throttling mode, in which the throttle interval is not included in the NOTIFY request. However, if a **throttle** parameter is specified in the SUBSCRIBE request, the throttle interval is included in the NOTIFY request.

The default mode is configured separately per event package and applies only for the event packages for which it was enabled. In particular, this mode is configured separately for the **presence**, **presence.winfo** and **presence.winfo.winfo** event packages.

Minimum and Maximum Intervals

Presence Server imposes minimum and maximum throttle intervals. The purpose of setting a minimum and maximum is to ensure that the server performance is not harmed. An interval that is too small (for example, 5 seconds) creates overhead on the server, without a real benefit for the client. In the case of an interval that is too big (for example, 15 minutes), it would probably make more sense for the client to send a fetch request every 15 minutes, and save the overhead (especially memory consumption) of keeping an idle subscription for such long times.

Both minimum and maximum intervals are configurable. The defaults are 5 seconds minimum and 10 minutes maximum. When a subscription request specifies a throttling interval that is outside of the configured range, the server sets the throttle to the minimum value or maximum value as appropriate. The adjusted throttle value is included in the Subscription-State header field's **throttle** parameter in each of the NOTIFY messages sent to the subscriber.

The minimum and maximum intervals are configured separately for each event package. If the throttle interval exceeds the subscription expiration value, Presence Server does not change the interval in the response to the subscriber. In this case, the effective interval is until the subscription expires.

Partial State Notification

When a subscriber has requested to receive partial notifications, Presence Server buffers all patch operations from all partial notifications that were throttled during the throttle interval. At the end of the throttle interval, all buffered patch operations are composed into one pidf-diff document. The composition is done by appending all patch operations together. If the result pidf-diff document is larger than the full presence document, the server sends the full state instead

Watcher Information Subscriptions

Watcher information subscriptions can also be throttled. If the state of at least one watcher has changed during the throttle interval, Presence Server sends a partial `<watcherinfo>` document at the end of the period, containing only the watchers whose state has changed.

Presence data storage

IBM WebSphere Presence Server Component stores required information about incoming subscriptions and for managing subscriptions to external providers.

Presence server uses multiple databases including:

1. PSDB - The main Presence Server where presence documents are stored
2. PSCIDB - The content indirection database for storing notification documents created for requests using content indirection
3. PSURDB - Presence server usage record database.
4. PSSIBUS (Optional) - Used if you are using the data store option for JMS messaging

Presence Server stores all presence information published by clients and also the presence information from external sources in the database **Publish** table. Presence Server aggregates and composes all published presence information into the presence document, that Presence Server stores in the separate database table: **Full Document**.

If watcher information (`watcherInfo`) is enabled,Presence Server also stores watcher information in the **Watchers** table in the database.

If authorization rules (`presenceRules`) are enabled, Presence Server stores presence rules for the subscribed users in **Presence Rules** table.

If white and black lists (`authorizationLists`) are is enabled , Presence Server stores the white/black list in the memory

Presence Server generates a usage record for each PUBLISH, SUBSCRIBE, and NOTIFY event. Usage records are stored in a database and contain information that describes how the service was used. Each usage record contains common event data that can be used to uniquely identify each record, and includes application-specific attributes. Usage records describe how requests end: either successfully or with an error. You can also use them for charging functions.

The Presence data model

IBM WebSphere Presence Server Component conforms to the Presence data model standard.

According to the data model, specified in IETF RFC 4479, each presentity has only one document, and–if possible–each document should have only one `<person>` element. Therefore, a presence composer should merge multiple `<person>` elements coming from different sources into one `<person>` element. Similarly, there should be only one `<device>` element per physical device and only one `<tuple>` element for a given contact, and the presence server should merge these elements too.

The RFC also stipulates that the presence composer may not always be able to merge information from multiple sources if ambiguities exist. Although a presence composer may be able to resolve ambiguities in some cases, in most cases it is best left to the watcher to resolve the ambiguity, by applying application-specific logic. Therefore, Presence Server does not try to resolve ambiguity. If information from multiple sources is ambiguous, that is, may contain conflicting information, Presence Server does not merge the conflicting `<person>`, `<tuple>`, or `<device>` elements. Instead, it keeps the original elements as distinct occurrences of the same element. The decision whether to merge is done separately per element. For example, Presence Server might merge the multiple occurrences of the `<device>` elements of device *A* but keep the distinct occurrences of `<device>` elements of device *B* because of ambiguity.

Presence Server attempts to merge all <person> elements, all <device> elements that have the same <deviceId> child element, and all <tuple> elements that have the same <contact> child element. <contact> elements are considered the same if the contact itself is the same, and the Priority attribute (if it exists) is the same.

The decision as to whether the elements can be merged is done by comparing the child elements. If there is no overlap between the child element types, that is, each of the merge candidates has different types of child elements, Presence Server can merge the elements. If there is an overlap, that is, child elements of the same type exists in more than one of the merge candidates, Presence Server merges only if these child elements are equal. Equality of child elements follows the definition in the document object model (DOM) and tested with the DOM isEqualNode method. In short, nodes of the same element type are considered equal if they have the same attribute key-value set (regardless of the order of the attributes within the element tag), and the child nodes are equal, that is, same number of child nodes and contain equal nodes at the same index.

As an example, consider these two <tuple> elements:

```
<tuple id="eg92n8">
      <contact priority="1.0">
              mailto:someone@example.com
      </contact>
      <status>
              <basic>open</basic>
      </status>
      <rpid:status-icon>
              http://example.com/mail.png
      </rpid:status-icon>
</tuple>
<tuple id="hgy7y1">
      <contact priority="1.0">
              mailto:someone@example.com
      </contact>
      <status>
              <basic>open</basic>
      </status>
      <rpid:class>email</rpid:class>
</tuple>
```

These two tuples can be merged. They have the same contact and contact priority. They have one overlapping child element, the <status> element, but the two <status> elements are equal. The merged element contains the <status-icon> element from the first tuple, and the <class> element from the second one:

```
<tuple id="eg92n8">
      <contact priority="1.0">
              mailto:someone@example.com
      </contact>
      <status>
              <basic>open</basic>
      </status>
      <rpid:status-icon>
              http://example.com/mail.png
      </rpid:status-icon>
      <rpid:class>email</rpid:class>
</tuple>
```

When comparing <note> elements appearing within <tuple>, <person> or <device> elements, the language (lang) attribute is taken into consideration. Note that elements having different lang attributes are not considered conflicting, and can be merged.

According to the Rich Presence Information Data (RPID) and Presence Information Data Format (PIDF) schemas, no child elements of <person>, <tuple>, or <device> can occur more than once (except for <note>). However, custom elements may be added that do occur more than once. When comparing the elements in this case, the comparison of child elements that occur more than once is done with the order preserved.

Interactions with other IBM WebSphere software for Telecom

IBM WebSphere Presence Server Component interacts with the other IBM WebSphere software for Telecom products and supports certain IMS reference points.

Presence Server uses IBM WebSphere XML Document Management Server Component to get both public and user defined lists, such as buddy lists. IBM XDMS acts as a repository for the XML documents and provides Presence Server access to these documents.

Presence Server can interact with a network agent to get information from the environment through the SIP methods. The following reference points, which are defined in the ETSI 123.141 standard, are supported:

- Pen
- Pex

Conformance with industry standards

Presence Server adheres to industry standards and uses industry standard protocols for easy deployment and integration with existing applications.

Presence information is sent and received using the SIP and HTTP protocols. It deploys on WebSphere Application Server and utilizes the SIP container included with WebSphere Application Server. It is fully integrated into the J2EE architecture. It uses standard JDBC data access methods for easy integration with IBM DB2® Enterprise Server Edition and Oracle Database JDBC compliant databases.

Chapter 2. Planning for IBM WebSphere Presence Server Component

Before you install IBM WebSphere Presence Server Component, it is important to understand how Presence Server integrates with other elements in the network (for example, IBM WebSphere XML Document Management Server Component), the hardware topologies on which it is typically deployed, the scaling methods that are available, and considerations for security.

Hardware and software requirements

Specific hardware and software is required before you can begin the installation process.

Hardware requirements

Hardware requirements vary, depending on the operating system on which you plan to deploy the IBM WebSphere software for Telecom products.

Before you begin the installation, one of the following operating-system platforms must be installed and configured. Choose a platform to display a detailed list of hardware requirements.

"AIX"

"Linux on PowerPC"

"Linux on Intel" on page 20

This information represents the minimum requirements. For greater performance and scalability, additional hardware may be needed.

AIX®

Processor
Power 5, 1.5 GHz, 32- and 64-bit

Physical memory
4 GB minimum, 2 GB per JVM recommended

Disk space
2 GB of free space (minimal)

4 GB of free space recommended

Other: CD-ROM or access to shared network drive where CD images are available

Linux® on PowerPC®

Processor
Power 5, 1.5 GHz, 32- and 64-bit

L2 cache
L2 cache for 2.8 GHz processor must be 512 KB

L2 cache for 3.4 GHz processor must be 1 M

Physical memory
4 GB minimum, 2 GB per JVM recommended

Disk space
2 GB of free space (minimal)

4 GB of free space recommended

Linux on Intel®

The following configuration is supported for Intel x86 platforms:

Processor
Pentium® 4, a minimum of 2 processors is required

2.8 GHz (32- and 64-bit)

L2 cache
L2 cache for 2.8 GHz processor must be 512 KB

L2 cache for 3.4 GHz processor must be 1 M

Physical memory
4 GB minimum, 2 GB per JVM recommended

Disk space
2 GB of free space (minimal)

4 GB of free space recommended

Other CD-ROM or access to shared network drive where CD images are available

Hyper-threading should be enabled

Software requirements

Required software includes the operating system, the WebSphere Application Server Network Deployment product (also referred to as WebSphere Application Server), Java, and a database component. Some Presence Server features also require ANT.

The information provided here is intended for a basic installation that is not scaled or fully deployed.

The following software should be installed and configured before you begin the installation process:

"Operating systems"

"Application servers" on page 21

"Java version" on page 21

"Databases" on page 21

ANT

Operating systems

The following operating systems are supported:

- Red Hat Enterprise Linux AS 5.0 Update 2
- SUSE Linux Enterprise Server 10 SP1
- AIX 5L 5.3 TL 07 04-0818

Application servers

One of the following application server offerings is required:

- WebSphere Application Server Network Deployment, version 7.0.0.1
- WebSphere Application Server Network Deployment, version 6.1.0.21

For a list of required WebSphere Application Server fixes, refer to the readme file, `WebSphereSoftwareForTelecomReadme.html`, on the QuickStart CD.

Java™ version

The following JDK versions are required:

- WebSphere Application Server version 7.0.0.1 requires JDK version 1.6.0 SR 3.
- WebSphere Application Server version 6.1.0.21 requires JDK version 1.5.0 SR 8.

Note: Your installation of WebSphere Application Server includes the correct JDK.

Databases

Each component has different database needs. Refer to the planning section for each component to understand the database needs for that component.

The following databases are supported:

IBM DB2 Enterprise Server Edition, version 9.5 FixPak 1

Oracle Database, version 10.2.0.4, 10.2.0.6, or 11.1.0.7

ANT

ANT must be installed on the server where you configure IBM WebSphere Presence Server Component. The configuration process for Presence Server invokes an ANT script.

You can obtain ANT from the Apache Ant Web site: http://ant.apache.org.

Planning to install IBM WebSphere Presence Server Component

Presence Server is deployed in a WebSphere cluster and uses databases to store important application information.

Application server considerations

Presence Server is deployed as an enterprise application on WebSphere Application Server, in a WebSphere clustered environment that consists of a SIP proxy server and a cluster containing one or more SIP application servers. The SIP proxy server must be defined and associated with the cluster.

Refer to the topic *Evaluating your hardware environment* for more information about the various ways in which you can set up the clusters.

Database considerations

Presence Server uses the following databases. Some of the databases are optional, depending on your configuration.

Table 1. Presence Server databases

Database name	Description
PSDB	Stores the Presence Server tables and is the main application database.
PSURDB	Stores usage records.
PSCIDB	Used to store content indirection documents. Optionally, the tables and records for this database could be placed in the *PSDB* database instead.
PSSIBUS	Optional; required only if you are using the Data store option for JMS messaging, instead of File store.

Evaluating your hardware environment

Presence Server installs and runs as an application on WebSphere Application Server. It can be deployed on various hardware configurations.

You can designate a separate resource list server (RLS) component to handle all subscriptions on Presence lists requests, and you can set up a routing scheme in which different Presence Server clusters handle requests from different sets of users.

WebSphere Application Server supports numerous deployment topologies. It is beyond the scope of this documentation to provide detailed steps for each topology. Therefore deployment information has been grouped into a number of broad categories. Throughout the documentation the categories are used to provide a reference point. Each component has a unique deployment strategy. Prior to deployment, review all of the planning and installation information.

Here is a list of the most commonly used topologies in a WebSphere Application Server environment:

Note: The single server topology can be used for development or the proof of concept.

Note: For Presence Server, only clustered environments are supported. However, you can create a cluster with only one member.

Vertical scaling topology
Members of a cluster exist on the same physical machine. Some services perform better with a small or moderate size Java heap. This may not utilize all of the resources of a powerful machine, so a vertically scaled deployment allows the processor and memory to be more fully utilized, while each instance can run more efficiently in a smaller JVM heap.

Frequently, vertical scaling is combined with horizontal scaling to allow both the efficient use of resources and the benefits of physical redundancy.

Horizontal scaling topology
Members of a cluster exist on multiple physical machines, effectively and efficiently distributing the workload of a single instance. Clustering is most effective in environments that use horizontal scaling because of the ability to build in redundancy and failover, to easily add new horizontal cluster

members to increase capacity, and to improve scalability by adding heterogeneous systems into the cluster.

You can combine vertical and horizontal scaling techniques to increase efficiency in the environment.

The database is shared and clustered.

Development topology

An IBM WebSphere Telecom Toolkit development environment can help you rapidly develop and deploy applications. This toolkit is available as a free download. It is designed to reduce the time to develop applications that use the Presence Server and other IBM WebSphere software for Telecom program products. The toolkit includes:

- The Presence Server API JAR file
- A REST client sample application that demonstrates the Presence Server REST APIs
- A SIP external source sample application (SIPp script) that simulates a SIP external source server
- A set of code snippets for you to use in developing your applications.

Integrating with supported network elements

Presence Server interacts with a group list server, using several of the features in IBM WebSphere XML Document Management Server Component (IBM XDMS). It also uses SIP external sources for retrieving presence information. For example, the Serving-Call/Session Control Function (S-CSCF) acts as an external registrar for retrieving registration information about users.

S-CSCF

Presence Server uses the Serving-Call/Session Control Function (S-CSCF) as a registrar. The registrar tracks where a user can be contacted and provides that information to callers.

S-CSCF is one possible source for SIP external data, which can be used to provide a rich source of registration information about users.

For details about setting up your configuration, see the topic *Configuring Presence Server to interact with S-CSCF sources*.

SIP external sources

Presence Server also interacts with other SIP external sources, for example external communities.

For details about setting up your configuration, see the topic *Configuring for SIP external sources*.

IBM XDMS

Presence Server uses the XML XCAP and SIP as the communication mechanisms with a group list server, such as IBM XDMS, for SIP SUBSCRIBE and NOTIFY messages. IBM XDMS is used for storing group lists (resource lists), for retrieving authorization lists (black and white lists), and for getting presence rules documents.

Important: You must install Presence Server and IBM XDMS on the separate physical servers.

For details about setting up your configuration, see the topics *Configuring for integration with IBM XMDS server* and *Configuring authorization*.

Planning for the databases

There are multiple databases for Presence Server They can be installed on the same server as Presence Server, or they can be installed on different servers.

The following table describes the databases that are used with Presence Server

Note: Some databases are optional, based on your configuration.

Table 2. Presence Server databases

Database name	Description
PSDB	Stores the Presence Server tables and is the main application database.
PSURDB	Stores usage records.
PSCIDB	Used to store content indirection documents. Optionally, the tables and records for this database could be placed in the *PSDB* database instead.
PSSIBUS	Optional; required only if you are using the Data store option for JMS messaging, instead of File store.

During the installation process, an XML file is used to capture your specific configuration information. The configuration information is then written to the Presence Server database, where it can be shared by all servers in the cluster.

The Presence Server database contains the following tables:

Table 3. Presence Server database

Table	Description
CONFIG	Stores configuration information
PUBLISH	Stores raw presence documents
FULLDOC	Stores combined presence documents
MANAGING_SERVERS	Stores information about the managing servers for subscriptions
WATCHERS	Stores pending subscription requests for watcher information
PRESENCE_AUTH_RULES	Stores presence rules documents

Table 4. Content indirection database

Table	Description
CONTENT_INDIRECTION	Stores content indirection document information

Tip: Optional: The CONTENT_INDIRECTION table could also be located in the Presence Server database.

Table 5. Usage records database

Table	Description
USAGERECORDS	Stores usage record information

The SiBus database contains multiple tables created by WebSphere Application Server for the SIBus function.

Planning for high availability

For high availability–including scalability, load balancing, and failover–you can configure WebSphere Application Server so that Presence Server runs in vertical and horizontal clusters.

A cluster is a group of servers that are managed together and participate in workload management. SIP and HTTP (for REST) proxy servers are used to route requests among the servers in the cluster.

Because Presence Server is deployed in a clustered configuration, it offers the following potential benefits:

- Scalability: Use vertical clustering to take full advantage of the resources of a multiprocessor system. Use horizontal cloning to allow for future growth.
- Load balancing: Reduce bottlenecks and other performance issues by distributing and balancing the workload of a single server across multiple servers.
- Failover: When one server fails, you can activate the sessions from the failed server on another server in the cluster so that normal operations proceed without interruption.

Planning security for Presence Server

IBM WebSphere Presence Server Component uses various methodologies to provide user and message security. WebSphere Application Server Network Deployment application security should be enabled on all servers that are running Presence Server.

General considerations for setting up security for Presence Server

Refer to these general considerations to achieve the most efficient and secure use of Presence Server. It is assumed that WebSphere Application Server Network Deployment application security is enabled when you run Presence Server.

The authorization APIs

Presence Server provides APIs with which you can develop customized authorization policies. Developers can use the authorization APIs to write pluggable applications that provide authorization information. This can be an effective way to increase the level of user privacy control for users.

An external authorization program registers with the APIs to manage authorization on a specific event package. The authorization APIs are used only when presence authorization rules are disabled and when white lists and black lists are also disabled.

For more information about the authorization APIs, refer to the topic *Extending Presence Server authorization*.

For more information about developing presence authorization rules, refer to IETF RFC 4745 and RFC 5025.

Increasing the level of user privacy by integrating with IBM WebSphere XML Document Management Server Component

You can also increase the level of user privacy control for users by integrating with IBM XDMS.

Presence Server uses the XML XCAP and SIP as the communication mechanisms with a group list server, such as IBM XDMS, for SIP SUBSCRIBE and NOTIFY messages. IBM XDMS is used for storing group lists (resource lists), for retrieving authorization lists (black and white lists), and for getting presence rules documents.

For details, see the topic *Configuring for integration with IBM XDMS*.

Authentication for SIP requests

Presence Server uses the Trust Association Interceptor (TAI) security component, included with the WebSphere IMS™ Connector product, to handle authentication for incoming SIP requests.

Presence Server depends on the Call Session Control function (CSCF) to check the user's authentication and eligibility for the service before the request reaches Presence Server. Trust Association Interceptor takes the user identity from the p-asserted-identity header that is added to the request by the CSCF and creates a WebSphere Application Server principal.

To authorize incoming SIP requests, the TAI inserts the WebSphere Application Server principal value into every request. Presence Server uses the asserted identity header on each incoming request, and it uses this identity for all user and request authorization. Presence Server can be configured to use the previous identity in case the p-asserted-identity header is omitted in subsequent requests.

For more information about how Presence Server uses TAI security, refer to the topic *Planning authentication security using the Trust Association Interceptor*.

The authentication mechanisms in Presence Server are designed to support content indirection, in which a SIP message can contain an indirect reference to desired content. See the topic *Content indirection* for more information.

Authentication for HTTP requests

Presence Server does not perform authentication for incoming HTTP requests. Instead, when a request reaches Presence Server, it is assumed that the request has already been authenticated by another component. It is further assumed that an

authenticated identity exists in the WebSphere Application Server principal object. This is the identity that Presence Server uses for authorization.

In a typical deployment, an IMS proxy component authenticates the sender of the request and adds an X-3GPP-Asserted-Identity header. The IMS HTTP interceptor then extracts this header and translates it to a WebSphere Application Server principal, similar to the way in which the P-Asserted-Identity header is handled for SIP. Because Presence Server relies solely on the WebSphere Application Server principal, there is no dependency on the Trust Association Interceptor.

A non-IMS or WebSphere-only deployment is also supported for HTTP requests. In such a deployment, another authentication method, such as base WebSphere authentication, can be used.

As is the case with SIP requests, unauthenticated requests may be accepted if they are allowed by the presence rules and if they are not specifically blocked in the server configuration.

White list and black list authorization

Presence Server uses two lists of identities (URIs) for authorization. Identities in the *white list* are always authorized. Identities in the *black list* are never authorized. The lists are stored by the IBM XDMS component as documents in the format specified by the mime type `application/resource-lists+xml` in IETF RFC 4826 (XML formats for representing resource lists).

On startup, Presence Server retrieves the white and black lists from IBM XDMS by means of an XCAP GET operation. Presence Server also subscribes to changes in the lists. When changes occur they affect new, incoming SIP requests, but they are not applied retroactively to existing SIP dialogs. Presence Server will apply the new authorization list for each user on all new and subsequent requests.

Authorizing incoming requests

Presence Server applies certain guidelines when determining whether to authorize incoming SUBSCRIBE and PUBLISH requests. Presentities (users) can define presence authorization rules to control who can access their presence information and what parts of the information are exposed. (For more information, refer to the topic *Presence authorization rules*.)

The following guidelines apply to both SIP requests and HTTP requests.

Here are the guidelines by which Presence Server authorizes SUBSCRIBE requests:

- White-list subscribers are always allowed to subscribe to all presentities. This overrides presence authorization rules.
- Black-list subscribers are never allowed to subscribe. Again, this overrides presence authorization rules.
- When a subscriber is not specified in either list, presence authorization rules are applied.

Here are the guidelines by which Presence Server authorizes requests to subscribe to watcher information:

- White-list subscribers are always allowed to subscribe.
- Only white-list subscribers are allowed to subscribe on `group/uri-list` for watcher information.

- Black-list subscribers are never allowed to subscribe.
- When a subscriber is not specified in either list, Presence Server verifies that the identity of the authenticated subscriber matches the watched identity as indicated in the ″To″ header of the SIP message. This guarantees that a user who is not on the white list can subscribe only to his or her own watcher information.

Here are the guidelines by which Presence Server authorizes PUBLISH requests:

- White-list publishers are always allowed to publish and to enable administrator publication on behalf of other users.
- Black-list publishers are never allowed to publish.
- When a publisher is not specified in either list, Presence Server verifies that the identity of the authenticated publisher matches the published presentity as indicated in the ″To″ header of the SIP message. This guarantees that a user who is not on the white list can publish his or her own presence information.

For all PUBLISH requests, Presence Server also verifies that the presence identity appearing in the PIDF document matches the identity indicated in the ″To″ header. When the identities do not match, the PUBLISH request is rejected–even if the publisher is on the white list.

Handling anonymous users

Presence Server treats anonymous users as special identities. To block anonymous access, it is recommended that you put the `Anonymous` identity in the black list. You can specify presence authorization rules for anonymous users. As a result, presence authorization rules are evaluated when a subscriber is found to be anonymous.

Message security

Presence Server supports the Transport Layer Security (TLS) protocol for SIP messages, as defined in IETF RFC 3261. RFC 3261 requires the URI scheme for SIP over TLS to use the `sips:` prefix; for example, `sips:johndoe@example.com`. Using TLS preserves the confidentiality and integrity of messaging, provides authentication and privacy of the participants in a session, and prevents denial-of-service attacks.

Presence Server adds a p-asserted-identity when it works with external registrars and with the IBM XDMS. The header, which describes the identity of the Presence Server application, is added to the first SUBSCRIBE request to each external registrar, to the IBM XDMS, and to each SIP external server.

Planning authentication security using the Trust Association Interceptor

The Trust Association Interceptor security component is intended to enhance the overall authentication security for the IBM WebSphere software for Telecom. An implementation scenario describes how you can deploy the TAI for Presence Server.

About the scenario

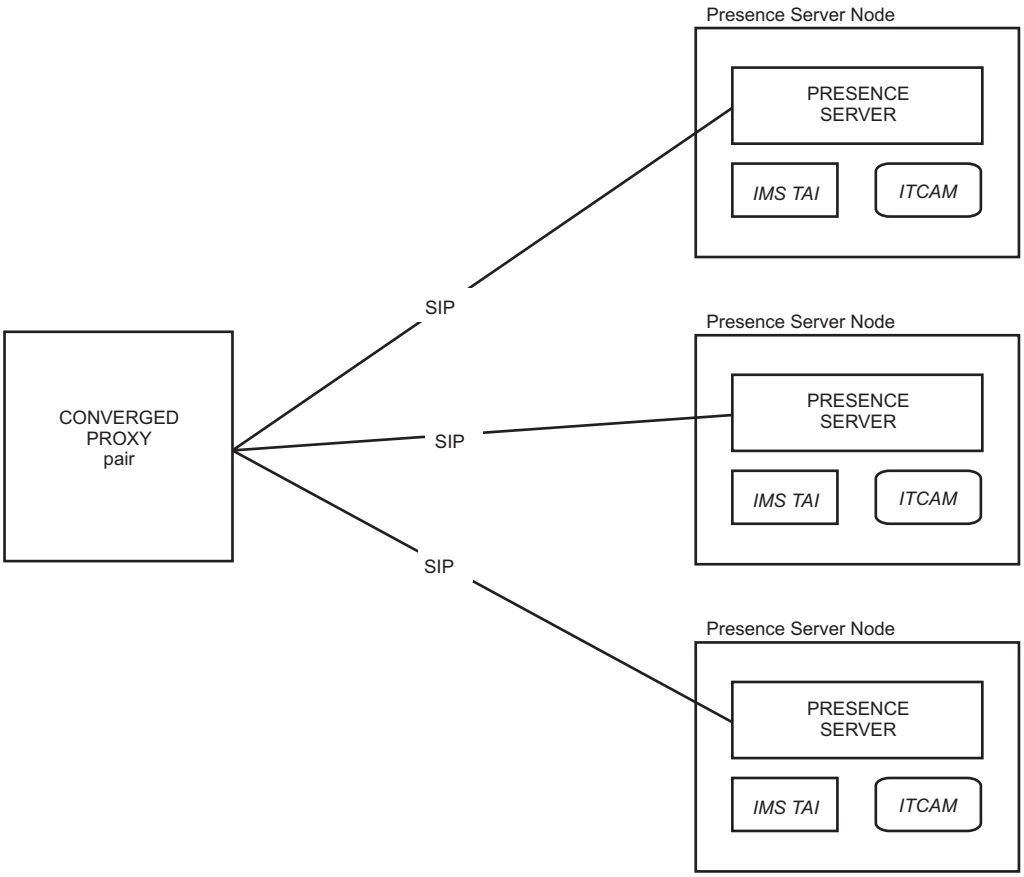
The following section depicts a common system configuration in which components are deployed in a production scenarios. The scenario is presented with the following conditions:

- The scenario does not illustrate all of the possible valid combinations of the IBM WebSphere software for Telecom.
- WebSphere Application Server Administrative node deployment is not shown. It is assumed that all components described in this section belonging in a cluster also belong to a WebSphere Application Server-administered core group for purposes of administration and management.
- While the scenario depicts the standard deployment of the IBM Tivoli Composite Application Management (ITCAM) for J2EE operations component that is co-located with the IBM WebSphere software for Telecom, the required remote ITCAM for J2EE components (or other SNMP-based monitors) are not shown. In all cases for interaction with the ITCAM for J2EE operations performance Servlet, it is expected that the cluster load balancer or proxy is not invoked to route requests.
- Session data replication details are not shown.
- The configuration does not address high-availability in an end-to-end sense, nor does it address interactions with required databases.
- Issues of development-to-deployment using the IBM WebSphere Telecom Toolkit are not addressed.
- The example illustrates the components and nodes in a cluster, in the WebSphere Application Server scaling sense consisting of two or more nodes providing identical service. These nodes may or may not cooperate at some level to guarantee high availability. Three nodes are shown as a convention, but other configurations are possible.

  Note: The Network Deployment high availability (HA) schemes require an even number of nodes in order to support pair-wise replication

Presence Server implementation scenario

The following diagram illustrates the scenario. Three Presence Server nodes, with the Trust Association Interceptor deployed on each one, receive SIP traffic that flows through a converged proxy.



Note:

- The WebSphere Application Server converged proxy or any third-party load balancer may be deployed pair-wise (for HA reasons).
- The converged proxy must be used for SIP traffic to maintain session affinity.
- The Trust Association Interceptor detects authenticated user identity from inbound messages.
- IBM Tivoli Composite Application Management (ITCAM) users communicate directly with the Presence Server node for purposes of collecting PMI data from that node.
- The S-CSCF is the reverse proxy security server (RPSS) and performs authentication.
- The S-CSCF adds a P-asserted identity header to the SIP message.

Planning to migrate from a previous release of IBM WebSphere Presence Server Component

To migrate the Presence Server application from version 6.2 to version 7.0, begin by creating a new cluster for the version 7.0 application instance. The new cluster runs in parallel with the existing cluster until a suitable amount of system activity has made the transition to the new cluster.

None of the Presence Server resources (data sources, topics, thread pools, sessions, and so forth) are shared between the version 6.2 and version 7.0 application instances. As a result, each installation functions independently while the migration is in progress.

The following assumptions and prerequisites apply for the migration process:

- This process assumes an "in-place" migration, with no new hardware resources being added.
- Failover is not supported during the migration period.
- The migration is complete when all operations are running on the new version 7.0 cluster. You have the option of either keeping the version 6.2 cluster running until all sessions have ended, or stopping the cluster when you believe that a suitable threshold of activity has been reached.
- Sessions can be several hours in length and can be refreshed by the client before expiring. Because of this, new dialogs (which can be routed to the new version 7.0 cluster) are created only when clients are switched on.

Database considerations

Presence Server version 7.0 uses a different database schema than it used in version 6.2. As a result, you will create new databases for your version 7.0 installation rather than retaining the version 6.2 databases.

Here are some general considerations for migrating the data in your version 6.2 Presence Server database:

- The installation package contains scripts (`db2Get` and `oracleGet`) with which you can extract your existing configuration settings from the database and store them in a temporary file. Then, after you install Presence Server version 7.0, you can copy the configuration settings to the new `SystemConfiguration.xml` file. From there, they are loaded into the new database.
- User information is not retained during the migration from a version 6.2 system to a version 7.0 system. (User information includes information published by users, information about watchers, information about managing servers, and information about documents for which content indirection is supported.) As the migration proceeds, new operations–for example, publish and subscribe requests–are initiated on the version 7.0 cluster, and the data associated with those operations is recorded in the new version 7.0 database. Note that watcher history information (information about watchers that are in waiting state, with no active subscriptions) is stored only in the version 6.2 database and is not retained.
- Usage records are not retained during the migration. If you need to preserve usage records after the migration, you should plan to extract them from the version 6.2 database after all activity has ended on the version 6.2 cluster.
- Because some persistence data is located in the version 6.2 database and some is located in the version 7.0 database, failover is not supported during the migration period.

Coexistence with other components

Presence Server version 6.2 can coexist with IBM WebSphere XML Document Management Server Component (IBM XDMS) version 7.0 for group list and presence-rules operations. Presence Server version 6.2 uses the `ua-profile` event package to subscribe to IBM XDMS in order to receive group structure. As long as

XDMS groups are created with no local lists, Presence Server version 6.2 will be able to manage and handle the XDMS group structure.

Presence Server version 7.0 can coexist with IBM XDMS version 6.2. A configuration setting, `enableXcapEvent`, controls whether Presence Server uses the `ua-profile` event package, which is compatible with IBM XDMS version 6.2. Note that when the `enableXcapEvent` setting is disabled, Presence Server version 7.0 does not support resource-list services documents and local lists in particular.

Chapter 3. Installing IBM WebSphere Presence Server Component

IBM WebSphere Presence Server Component supports a clustered installation configuration. With a clustered environment, you can have a cluster of one or many cluster members.

This installation scenario requires you to installWebSphere Application Server Network Deployment with a cell environment, which creates the deployment manager and a federated node. If your WebSphere Application Server Network Deployment instance does not include a cell environment, use the Profile Management Tool to create a new profile. When creating the new profile, select the **Cell (deployment manager and a managed node)** option. Other options are available for setting up a cell environment. Refer to the WebSphere Application Server documentation for more information.

Migrating from a previous release of IBM WebSphere Presence Server Component

For users of previous versions of the Presence Server product, the topics in this section describe the process of migrating your existing configuration to the current version.

Before you begin

Deploy WebSphere Application Server Network Deployment version 7.0.0.1 or 6.1.0.21.

ANT must be installed on the server where you configure IBM WebSphere Presence Server Component. You can obtain ANT from the Apache Ant Web site: http://ant.apache.org.

About this task

When migrating from a previous version of Presence Server, perform the following steps:

1. Create new database instances using the instructions in the topic *Preparing the databases*.
2. Copy your version 6.2 configuration data to a temporary file, using one of the scripts (`db2Get` or `oracleGet`) that are provided on the installation medium:

   DB2
   ```
   ant db2Get -f GetPresenceConfiguration.xml
   -DdbUserName=db_user -DdbUserPassword=db_user_pw -DdbServer=hostname
   -DdbName=db_name -DdbPort=port -DjdbcDir=drivers_path
   -DconfigurationFileName=dest_file_name
   ```

   Oracle
   ```
   ant oracleGet -f GetPresenceConfiguration.xml
   -DdbUserName=db_user -DdbUserPassword=db_user_pw -DdbServer=hostname
   -DdbName=db_name -DdbPort=port -DjdbcDir=drivers_path
   -DconfigurationFileName=dest_file_name
   ```

   where:

   db_user is the name of the database user.

db_user_pw is the password for the database user.

hostname is the fully qualified name of the database host.

db_name is the name of the database.

port is the port number through which the database is accessed.

drivers_path is the path to the JDBC drivers for the database.

dest_file_name is the directory path and name for the temporary file in which your configuration data will be stored.

3. Create a new cluster for Presence Server on the WebSphere instance. Refer to the topic *Creating the cluster* for details.
4. Install the Presence Server version 7.0 program code using the procedures in the topic *Installing the Presence Server product*.
5. Copy the configuration parameters you saved in step 2 on page 33 to the version 7.0 `SystemConfiguration.xml` file
6. Route all new SIP/HTTP requests to the new Presence Server version 7.0 proxy server. Old sessions will continue to be routed to the version 6.2 cluster. When long-running version 6.2 sessions end, they will be rerouted to the new version 7.0 cluster.
7. Monitor performance on the version 6.2 cluster. When the Presence Server indicators reach zero, or when you believe that a suitable threshold of activity has been reached, stop the Presence Server application on the version 6.2 cluster. The WebSphere Performance Monitoring Infrastructure (PMI) is recommended for monitoring performance. For details, refer to the topic *Monitoring system performance using WebSphere PMI*.
8. Uninstall the Presence Server version 6.2 program code using the procedures in the topic *Uninstalling IBM WebSphere Presence Server Component*.

Preparing the environment

Presence Server is deployed on a WebSphere Application Server cluster. A successful installation requires that the prerequisite software be installed and configured and that the server platform be configured properly.

Verifying that the required software is installed

Before installing IBM WebSphere Presence Server Component, you must install the prerequisite software.

About this task

To help you get started, use this procedure as a checklist to ensure you have the prerequisite software installed. This information is provided as guidance. For detailed instructions, refer to the product documentation for the prerequisite software.

If possible, use the default profiles for WebSphere Application Server Network Deployment and do not customize the profile with the Profile Management Tool. If you need to use an advanced profile, do not select the **for development** profile. It may cause problems with the WebSphere Application Server SIP container.

1. Verify that a supported DB2 or Oracle database server is installed. Supported database versions are listed in the topic *Software requirements*.
2. Verify that an appropriate version of WebSphere Application Server Network Deployment is installed. Supported versions are listed in the *Software requirements* topic in this information center.

When installing the application server, be sure to select the **Cell (deployment manager and a managed node)** option. If you are using a distributed topology where you have the deployment manager installed on a different server than the managed node, refer to the WebSphere Application Server Information Center for information about adding managed nodes.

3. Start the deployment manager. Run the following command:

   AIX `was_profile_root/bin/startManager.sh`

   Linux `was_profile_root/bin/startManager.sh`

   Where: *was_profile_root* is the WebSphere Application Server Network Deployment profile directory.

4. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.

   b. Enter an administrator user ID and password.
   c. Click **Log in**.
5. Verify that you have the correct version of WebSphere Application Server Network Deployment installed:
   a. Click **Welcome**.
   b. Click **WebSphere Application Server**.
   c. Under **About your WebSphere Application Server** , you should see a line of text indicating that your WebSphere version is one of those specified in the *Software requirements* topic.

What to do next

After you have installed the required software, you are ready to create the cluster or clusters on which you will deploy the Presence Server product.

Creating the cluster

To create the cluster, you can convert an existing application server and generate additional cluster members using the original cluster member as a template.

Before you begin

Review and verify that all hardware and software prerequisites have been met. Refer to the topics *Hardware requirements* and *Software requirements* for details.

In addition, you should have completed the following items:

- Installed WebSphere Application Server Network Deployment with a cell environment
- Created a deployment manager profile
- Created a federated node

1. Start the deployment manager. Run the following command:

AIX *was_profile_root*/bin/startManager.sh

Linux *was_profile_root*/bin/startManager.sh

Where:

The *was_profile_root* path contains the name of the deployment manager profile (for example, Dmgr01).

2. Start all node agents. Run the following command:

AIX *was_profile_root*/bin/startNode.sh

Linux *was_profile_root*/bin/startNode.sh

Where:

The *was_profile_root* path contains the name of a federated node profile (for example, Custom01).

3. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.

   b. Enter an administrator user ID and password.
   c. Click **Log in**.
4. In the navigation panel, click **Servers** → **Clusters** → **WebSphere Application Server clusters**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Clusters**.
5. Define the cluster:
   a. Click **New**.
   b. Type `WPS_cluster` for the **Cluster name**.
   c. Select **Configure HTTP session memory-to-memory replication**.
   d. Click **Next**.
6. Define the first cluster member:
   a. Type the server name for the **Member name**, for example `WPS_server1`.
   b. Click **Next**.
7. Add additional cluster members if necessary. Repeat the following steps for each cluster member you would like to add.
   a. Type the server name, for example `WPS_server2`. For federated nodes with multiple cluster members, you will need to know the names of each of the members.
   b. Select a node name from the drop-down list.
   c. Click **Add Member** to add the cluster member. The new member appears in the table of cluster members.
8. Click **Next**.
9. Click **Finish**.
10. Click **Save** to save changes to the master configuration.

11. Click **OK** when node synchronization has completed.

Defining the proxy server

You must create the proxy server.

Before you begin

Review and verify that all hardware and software prerequisites have been met. Refer to the topics *Hardware requirements* and *Software requirements* for details.

In addition, you should have completed the following items:

- Installed WebSphere Application Server Network Deployment with a cell environment
- Created a deployment manager profile
- Created a federated node
- Created the cluster

About this task

Ports are assigned to each proxy server automatically. Depending on your environment, you may need to change the port configuration of the proxy server. For additional information about configuring ports, refer to the WebSphere Application Server 7.0 Information Center.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.

   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. In the navigation panel, click **Servers** → **Server Types** → **WebSphere proxy servers**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Proxy Servers**.
3. Click **New**.
4. Create the proxy server:
   a. Select a node name from the drop-down list.
   b. Type `WPS_proxy` for the **Server Name**.
   c. Click **Next**.
   d. Click **Next** on the rest of the steps, accepting the default values, and click **Finish** on the last step.
   e. Click **Save** to save changes to the master configuration.
   f. Click **OK** when node synchronization has completed.

5. Click **Save** to save changes to the master configuration.

Preparing the Trust Association Interceptor for use

Follow these procedures to install the Trust Association Interceptor and make it ready for use by the IBM WebSphere Presence Server Component.

Configuring WebSphere security for the Trust Association Interceptor

Before you install the Trust Association Interceptor (TAI), you must configure WebSphere security to prepare it for use.

About this task

Follow these steps to configure WebSphere security so that the Trust Association Interceptor can be used with Presence Server.

1. On the deployment manager, log in to the WebSphere Integrated Solutions Console.
2. Click **Security** → **Global security** to display the Global security window.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Security** → **Secure administration, applications, and infrastructure**.
3. Use the Security Configuration wizard to enable security:
   a. Click **Security Configuration Wizard** to launch the wizard.
   b. Check **Enable application security**.
   c. Ensure that **Java 2 security** is *not* checked.
   d. Click **Next**.
4. Specify an administrative user:
   a. Click **Federated repositories**, then click **Next**.
   b. Type a valid user name and password into the text boxes.
   c. Click **Next**.
   d. Click **Finish**.
   e. Click **Save** to save changes to the master configuration.
   f. Click **OK** when node synchronization has completed.
5. Restart all WebSphere process in the cell (deployment manager, node agents and servers).
6. Verify the changes you made:
   a. Return to the Global security window.
   b. Click **Administrative User Roles**.
   c. Verify that the user name you defined is shown as the Primary Administrator user name.

Preparing the installation files for the Trust Association Interceptor

Before you install the Trust Association Interceptor (TAI), the WebSphere IMS Connector installation file must be unpacked on the servers where WebSphere Application Server is installed.

Before you begin

Unpacking the WebSphere IMS Connector installation file, `DHAImsConnectorInstallPackage_6.2.0.tar`, which is found on the WebSphere IMS Connector CD, places all of the files for the WebSphere IMS Connector and for the TAI into their appropriate directories.

It is necessary to perform this step only once, even if you plan to use the TAI for several different components.

Note: *was_root* is the installation root directory for WebSphere Application Server Network Deployment. By default, this directory is:

- AIX `/usr/IBM/WebSphere/AppServer`
- Linux `/opt/IBM/WebSphere/AppServer`

1. On the server where WebSphere Application Server is installed, copy the installation file (`IBM_WebSphere_IMS_Connector/DHAImsConnectorInstallPackage_6.2.0.tar`) from the CD to the *was_root* directory.
2. Change (`cd`) to the *was_root* directory.
3. Unpack the file by typing the following command: `tar -pxvf DHAImsConnectorInstallPackage_6.2.0.tar`

   Note: Remember to apply any relevant fix packs for the TAI.

Preparing the databases

You need to set up databases for storing Presence Server data.

For DB2 installations, the Presence Server installation medium contains scripts to help you define the databases.

For Oracle installations, you must create the databases manually.

For both database types, the required database tables are created later, when you install the Presence Server product.

Creating the Presence Server database for DB2

Before installing the Presence Server product, you must configure database for use by the application.

Before you begin

Before you begin, the following software should be installed:

- IBM DB2 Enterprise Server Edition, version 9.5 FixPak 1

1. Log in to the DB2 server as a database administrator.
2. Create a directory that has write and execute permission, for example `DB_dir`.
3. Copy the installation `.tar` file, `IBMPresenceServerDbPackage_7.0.tar`, from the installation medium to the new directory.
4. Switch to the new directory.
5. Unpack the installation `.tar` file using the following command:

   ```
   tar -xvf IBMPresenceServerDbPackage_7.0.tar
   ```

6. Switch to the following directory: *was_root*/installableApps/presence/scripts/dbScripts/presence.
7. Verify that the following files exist in /presence:
   - ConfigDB2.sh
   - RunConfigDB2.sh
   - CreateDB2Tables.ddl
8. Run the following commands to ensure that the database configuration scripts will run properly:

   ```
   chmod 755 ConfigDB2.sh
   chmod 755 RunConfigDB2.sh
   ```

9. Edit the database preparation script, RunConfigDB2.sh, for your environment:
   a. Open RunConfigDB2.sh using a text editor.
   b. Locate the line that begins with the following text:

      ```
      #Update command line parameters here.
      ```

   c. Update values for each parameter listed.

Parameter	Description	Default value
DBSERVER	Fully qualified name of the database server Linux Should match the Linux hostname command value if you are running the script directly on the database server	<hostname>
DBPORT	Number for the listening port on the database server	50000
DBNAME	Name of the database	PSDB
DBALIAS	Alias by which the database is known	PSDB
DBLOCALE	Territory code that identifies your DB2 locale, for example US or JP	US
DBINSTANCE	Name of the database instance	db2inst
DBINSTANCPW	Password for the database instance	<pw>
DBUSER	User ID for the authorized user (the ID with which you are logged in)	db2inst
DBUSERPW	Password for the authorized user	<pw>
DBDIR	Directory in which the database files are to be created (should be a directory with write permission)	/home/db2inst
DBCREATE	Boolean value specifying whether the database should be recreated	TRUE

For example:

```
DBSERVER=dbserver.example.com
DBPORT=50000
DBNAME=PSDB
DBALIAS=PSDB
DBLOCALE=US
DBINSTANCE=db2inst
DBINSTANCPW=inst_pw
DBUSER=db2inst
DBUSERPW=user_pw
DBDIR=/home/db2inst
DBCREATE=TRUE
```

d. Run the script.

10. Verify that the database was created properly by typing the following command, where *database_alias* is the alias you identified in the `RunConfigDB2.sh` script:

    `db2 connect to` *`database_alias`* `user` *`database_administrator_ID`*

    For example:

    ```
    db2 connect to PSDB user db2inst
    ```

    If the script ran properly, the following results display:

    ```
    Database Connection Information

     Database server        = DB2/LINUX 9.5.1
     SQL authorization ID   = database_administrator_ID
     Local database alias   = database_alias
    ```

Results

Note: If errors occur when you run the database preparation script, refer to the topic *Troubleshooting the database script*.

What to do next

The required DB2 database tables will be created for you when you run the Presence Server interactive installation or silent installation.

Creating the content indirection database for DB2

You must configure a content indirection database before installing Presence Server.

Before you begin

Before you begin, the following software should be installed:

- IBM DB2 Enterprise Server Edition, version 9.5 FixPak 1

1. Log in to the DB2 server as a database administrator.
2. Create a directory that has write and execute permission, for example `DB_dir`.
3. Copy the installation `.tar` file, `IBMPresenceServerDbPackage_7.0.tar`, from the installation medium to the new directory.
4. Switch to the new directory.
5. Unpack the installation `.tar` file using the following command:

   ```
   tar -xvf IBMPresenceServerDbPackage_7.0.tar
   ```

6. Switch to the following directory: *`was_root`*`/installableApps/presence/scripts/dbScripts/presence`.
7. Verify that the following files exist in `/presence`:
   - `ConfigDB2.sh`

- `RunConfigContentIndirectionDB2.sh`
- `CreateContentIndirectionDB2Tables.ddl`

8. Run the following commands to ensure that the database configuration scripts will run properly:

   ```
   chmod 755 ConfigDB2.sh
   chmod 755 RunConfigContentIndirectionDB2.sh
   ```

9. Edit the database preparation script, `RunConfigContentIndirectionDB2.sh`, for your environment:
   a. Open `RunConfigContentIndirectionDB2.sh` using a text editor.
   b. Locate the line that begins with the following text:

      ```
      #Update command line parameters here.
      ```

   c. Update values for each parameter listed.

Parameter	Description	Default value
DBSERVER	Fully qualified name of the database server Linux Should match the Linux `hostname` command value if you are running the script directly on the database server	<hostname>
DBPORT	Number for the listening port on the database server	50000
DBNAME	Name of the database	PSDB
DBALIAS	Alias by which the database is known	PSDB
DBLOCALE	Territory code that identifies your DB2 locale, for example US or JP	US
DBINSTANCE	Name of the database instance	db2inst
DBINSTANCPW	Password for the database instance	<pw>
DBUSER	User ID for the authorized user (the ID with which you are logged in)	db2inst
DBUSERPW	Password for the authorized user	<pw>
DBDIR	Directory in which the database files are to be created (should be a directory with write permission)	`/home/db2inst`
DBCREATE	Boolean value specifying whether the database should be recreated	TRUE

10. Run the `RunConfigContentIndirectionDB2.sh` script.
11. Verify that the database tables were created properly by typing the following command, where *database_alias* is the alias you provided in the `RunConfigContentIndirectionDB2.sh` script:

```
db2 connect to database_alias user database_administrator_ID
```

Where:

database_alias represents the database alias you assigned in step 7

database_administrator_ID represents the administrator user ID

For example:

```
db2 connect to PSICDB user db2inst
```

If the script ran properly, the following results display:

```
Database Connection Information

 Database server        = DB2/LINUX 9.5.1
 SQL authorization ID   = database_administrator_ID
 Local database alias   = database_alias
```

Results

Note: If errors occur when you run the database preparation script, refer to the topic *Troubleshooting the database script*.

What to do next

The required DB2 database tables will be created for you when you run the Presence Server interactive installation or silent installation.

Creating the usage records database for DB2

You must create a usage records database before proceeding with the installation. It is not necessary to define the database tables at this time: they will be defined later, when you run the Presence Server interactive installation or silent installation.

Before you begin

Before you begin, the following software should be installed:

- IBM DB2 Enterprise Server Edition, version 9.5 FixPak 1

1. Log in to the DB2 server as a database administrator.
2. Create a directory that has write and execute permission, for example `DB_dir`.
3. Copy the installation `.tar` file, `IBMPresenceServerDbPackage_7.0.tar`, from the installation medium to the new directory.
4. Switch to the new directory.
5. Unpack the installation `.tar` file using the following command:

   ```
   tar -xvf IBMPresenceServerDbPackage_7.0.tar
   ```
6. Switch to the following directory: *was_root*`/installableApps/presence/scripts/dbScripts/usageRecords`.
7. Verify that the following files exist in `/usageRecords`:
   - `crtsrvDB2.sh`
   - `UsageDbDB2.ddl`
8. Run the following command to ensure that the database configuration script will run properly:

   ```
   chmod 755 crtsrvDB2.sh
   ```
9. Launch the `crtsrvDb2` script by issuing the following command:

   ```
   ./crtsrvDb2.sh options
   ```

where: *options* corresponds to the following values.

Table 6. crtsrvDb2 configuration values

Parameter	Description	Recommended Value
dbServer	Fully qualified hostname for the database server.	myhost.example.com
dbPort	Number for the listening port on the database server.	50000
dbLocal	Boolean value specifying whether the database should be created in the local catalog (TRUE) or on a different server (FALSE).	TRUE
dbNodeName	Remote database server node name, when dbLocal is FALSE. This value is used for reference purposes and it is not the host name.	RDBSRV
dbName	Name of the database.	PSUR
dbAlias	Alias by which the database is known.	PSUR
dbLocale	Database locale or territory code, for example US or JP.	US
dbAdmin	Administrator user ID for the database instance.	db2inst1
dbAdminPW	Password for the administrator user ID.	<pw>
dbUser	User ID for accessing the database.	db2inst1
dbUserPW	Password for the database user ID.	<pw>
INITorDDLFile	One of the following: • INIT to initialize the database instance • Path to the DDL file that is used to define the database's attributes	INIT
infoflag	TRUE to display informational messages while the database is being initialized. FALSE to suppress informational messages.	TRUE

For example:

```
./crtsrvDb2.sh myhost.example.com 50000 TRUE RDBSRV PSUR PSUR US db2inst1 password db2inst1 pass
```

10. Verify that the database was created properly by typing the following command:

    db2 connect to *database_alias* user *database_administrator_ID*

    Where:

    database_alias is the alias you provided when running the crtsrvDB2.sh script

    database_administrator_ID represents the administrator user ID

    For example:

    ```
    db2 connect to PSUR user db2inst
    ```

    If the script ran properly, the following results display:

    ```
    Database Connection Information

     Database server        = DB2/LINUX 9.5.1
     SQL authorization ID   = database_administrator_ID
     Local database alias   = database_alias
    ```

Results

Note: If errors occur when you run the database preparation script, refer to the topic *Troubleshooting the database script*.

What to do next

The required DB2 database tables will be created for you when you run the Presence Server interactive installation or silent installation.

Creating the Service Integration Bus (SIBus) database for DB2

You can optionally configure a SIBus database before installing Presence Server.

Before you begin

Before you begin, the following software should be installed:

- IBM DB2 Enterprise Server Edition, version 9.5 FixPak 1

1. Log in to the DB2 server as a database administrator.
2. Create a directory that has write and execute permission, for example `DB_dir`.
3. Copy the installation `.tar` file, `IBMPresenceServerDbPackage_7.0.tar`, from the installation medium to the new directory.
4. Switch to the new directory.
5. Unpack the installation `.tar` file using the following command:
   ```
   tar -xvf IBMPresenceServerDbPackage_7.0.tar
   ```
6. Change (`cd`) to the following directory: *`was_root`*`/installableApps/presence/scripts/dbScripts/SIBus`.
7. Verify that the following files exist in `/SIBus`:
   - `createDB2SIBus.sh`
   - `dropDB2SIBusSchemas.sh`
8. Run the following command to ensure that the database configuration script will run properly:
   ```
   chmod 755 createDB2SIBus.sh
   ```
9. Issue the following command to create the database:
   `./createDB2SIBus.sh` *`db_name database_administrator_ID database_administrator_pw`*

   where:

   db_name is the name of the SIBus database, for example PSSIBUS

   database_administrator_ID is the authorized user for the database, for example `db2inst`

   database_administrator_pw is the password for the authorized user
10. Verify that the database was created properly by typing the following command:
    `db2 connect to` *`db_name`* `user` *`database_administrator_ID`*

    For example:
    ```
    db2 connect to PSSIBUS user db2inst
    ```

    If the script ran properly, the following results display:

```
Database Connection Information

 Database server        = DB2/LINUX 9.5.1
 SQL authorization ID   = database_administrator_ID
 Local database alias   = db_name
```

Results

Note: If errors occur when you run the database preparation script, refer to the topic *Troubleshooting the database script*.

What to do next

The required DB2 database tables will be created for you when you run the Presence Server interactive installation or silent installation.

Creating Presence Server databases for Oracle

It is necessary to create Oracle application databases before installing Presence Server.

Before you begin

Before you begin, the following software should be installed:

Oracle Database, version 10.2.0.4, 10.2.0.6, or 11.1.0.7

1. Log in to the Oracle server as a database administrator.
2. Manually create the databases that you need:
   - Presence Server application database: *PSDB*
   - Presence Server content indirection database: *PSCIDB*
   - Presence Server usage record database: *PSURDB*
   - (optional) Presence Server data store database: *PSSIBUS*

   Refer to the Oracle documentation for instructions on how to create the databases.

What to do next

The required Oracle database tables will be created for you when you run the Presence Server interactive installation or silent installation.

Installing the Presence Server product

You can install the IBM WebSphere Presence Server Component product either by using an interactive installer or by performing a silent installation.

Installing Presence Server using the interactive installer

Use the interactive installer to install the IBM WebSphere Presence Server Component.

Before you begin

- The Presence Server databases must be created on your database server.
- The database client for your database server must be installed in the same path on each Presence Server computer (each node).
- The WebSphere Application Server Network Deployment version should be 7.0.0.1 or 6.1.0.21.

- The WebSphere Application Server topology should be configured. For example, nodes should be created and configured in each cluster. A proxy server should also be created.
- WebSphere Application Server security should be configured with the preferred user repository.
- If you are using the WebSphere IMS Connector Trust Association Interceptor (TAI), the TAI JAR file should be installed on each Presence Server server.

About this task

For standalone configurations (one Presence Server instance in a cluster) this installation needs to be run only once on the standalone server. For clusters with managed nodes, the installation must be run on the deployment manager and each managed node. The installation is abbreviated when run on the managed nodes.

To install Presence Server, locate the installation setup file on the Presence Server CD ROM and follow these instructions:

1. Copy the `setup.bin` file from the CD-ROM to a directory on your system, and copy the `IBMPresenceServerDbPackage_7.0.tar` file to a directory on your database server.
2. Set the proper file permissions on the `setup.bin` file by typing `chmod 755 setup.bin` from the command prompt.
3. Launch the interactive installer. Run the following command: `./setup.bin`
4. Click **Next** at the introduction panel.
5. Select **I accept the terms of the License Agreement**, and click **Next**. The installer checks to verify that the correct version of WebSphere Application Server is installed and returns a list of qualified WebSphere Application Server installations.
6. Select a WebSphere Application Server installation and click **Next**.
7. Select the WebSphere Application Server profile to be used for the installation, and click **Next**. Example: **opt/IBM/WebSphere/AppServer/profiles/Dmgr01**
8. Select **Install**, and click **Next**.
9. Type the administrative user's name and password, and click **Next**. Example:

   WebSphere Administrator Name: `was_admin`

   WebSphere Administrator Password: `waspass`
10. Select the WebSphere scope where you are installing, and click **Next**. (For clustered installations, select the cluster you created during the prerequisite steps.)
11. Select a proxy server.
12. In a clustered installation, click **Configure CSCF Address** and type the address for the CSCF server. This is a semicolon-delimited list of hosts (IP addresses or host names) for which the proxy server will not remove the P-asserted identity headers that are added by the CSCF. The use of a mask (*) is permitted. Do not include spaces in the input string.

    This step is required is when you are installing or modifying a deployment manager on a cluster scope. It ensures that P-asserted identity headers that are added by the CSCF will not be removed by the proxy server. Examples of valid addresses: `192.0.2.21;*.*.*.*`
13. Click **Next**. The installer looks for previously installed versions of Presence Server and returns with the choice to modify or install. The instructions in the window apply whether you choose **Modify** or **Install**.

14. Click **Next**.
15. Optional: Select the **Configure TAI** check box, and type the list of allowed senders. This is a comma-delimited list of hosts (IP addresses or host names) that the interceptor considers to be trusted. Do not include spaces in the input string.

    If the IMS Trust Association Interceptor (TAI) is configured, this variable is used to set the allowed senders configuration parameter for the TAI. This step is required when you are installing or modifying a deployment manager. Additionally, the TAI JAR files must be installed on all nodes. The use of a mask (**x** for the TAI) is permitted.

    For information about allowed senders for the TAI, refer to the topic *Configuring the Trust Association Interceptor* in the IMS Connector portion of this information center. Examples of valid addresses:
    `x@us.example.com,x.example.com,192.2.x.x,x.x.x.x,2002:92A:8F7A:0:0:20:0:1`
16. Click **Next**.
17. Configure SIBus information by selecting **Data Store** or **File Store** and clicking **Next**.
18. Configure the database parameters
    a. Select **DB2** or **Oracle**.
    b. Click **Choose** and navigate to the directory where the JDBC JAR files are located. The JDBC JAR files should be installed in the location on all nodes.
    c. Click **Next**.
    d. Type the following information into the corresponding fields for the Presence Server database (*PSDB*):

Configuration parameter	Example
Database Host Name	`db2_server1`
Database Name	`PSDB`
Database Port Number	`5000`
Database Administrator Name	`db2inst`
Database Administrator Password	`db2password`

    e. Select the **Update Configuration Table** check box, and click **Next**.

    Note: Selecting this is necessary only if a previous configuration exists in the database. If the database is not already configured the configuration table will be updated anyway.

    f. Type the following information into the corresponding fields for the Presence Server content indirection database (*PSCIDB*), and click **Next**:

Configuration parameter	Example
Database Host Name	`db2_server1`
Database Name	`PSCIDB`
Database Port Number	`5000`
Database Administrator Name	`db2inst`
Database Administrator Password	`db2password`

    g. Click **Next**

h. Type the following information into the corresponding fields for the Presence Server usage record database (*PSURDB*):

Configuration parameter	Example
Database Host Name	`db2_server1`
Database Name	`PSURDB`
Database Port Number	`5000`
Database Administrator Name	`db2inst`
Database Administrator Password	`db2password`

i. Select the **Create Usage Record Tables** check box, and click **Next**.

Important: Select this check box only if you are using a new usage record database. Any previous information in the database will be removed if this is selected.

j. Optional: If you selected to use **Data Store** earlier, type the following information into the corresponding fields for the Presence Server SIBUS data store database (*PSSIBUS*):

Configuration parameter	Example
Database Host Name	`db2_server1`
Database Name	`PSSIBUS`
Database Port Number	`5000`
Database Administrator Name	`db2inst`
Database Administrator Password	`db2password`

Note: If you selected to use the File Store option, no additional configuration is required and this step is skipped.

k. Click **Next**.

19. Review the pre-installation summary, and click **Install**.
20. Review the installation summary.
21. Optional: To review the installation logs, navigate to *was_root*`/logs`.

Note: *was_root* is the installation root directory for WebSphere Application Server Network Deployment. By default, this directory is:

- AIX `/usr/IBM/WebSphere/AppServer`
- Linux `/opt/IBM/WebSphere/AppServer`

Results

The Presence Server application is ready to deploy. Use the primary configuration file, `SystemConfiguration.xml`, if you would like to modify the default configuration before deployment. This file is found in the directory *was_root*`/installableApps/presence/scripts/config`. (There is also a backup copy in *was_root*`/logs`.) For more information, refer to the *Configuring IBM WebSphere Presence Server Component* section of this information center.

Installing Presence Server silently

The Presence Server product can be installed silently without user interaction, by reading user responses from a response file.

A silent installation performs the installation of the product in silent mode, without the interactive installer. Instead of displaying the wizard interface, the silent installation allows the installation program to read all of your responses from a file that you provide.

To specify non-default options during a silent installation, customize the response file. To install silently, you must accept the license agreement using the agreement option within the response file.

The following topics describe the silent installation process.

Editing the response file

A silent installation uses a response file to inform the installer which actions to perform. This file must be customized before attempting a silent installation

About this task

Customizing the response file, `presenceServer_install.rsp`, involves setting the values of variables. The installer uses these variables during the silent installation process to determine what actions are required for the installation process.

Customize the response file precisely so that the installation program can read the option values accurately. For example, always enclose values in double quotation marks. When the installer encounters an incorrect parameter, it stops and writes an explanation of the problem to the installer log file: `presenceServerInstall.log`.

1. Open the installation response file, `presenceServer_install.rsp`, in a text editor. After the first installation, the response file can be found in the directory *`was_root`*`/installableApps/presence`.

   Note: As an alternative to the default response file, you can build a response file using the template found on the Presence Server CD in the `/response_files` directory: `presenceServer_install_template.rsp`. To use this file for silent installation, copy it to each cluster member and each deployment manager, then modify it.
2. Edit the response file, modifying the following values as needed. The following tables list the response file parameters.

   Note: Most of the response file parameters are used only on a deployment manager node or a stand-alone server. When you are installing on a clustered node that is not a deployment manager, only the following parameters are required:

   LICENSE_ACCEPTED

   MODE

   WAS_HOME

   WAS_PROFILE

Table 7. License and mode properties for silent installation

Property	Valid values	Example value
LICENSE_ACCEPTED=	Must equal true to use installer	*true*
MODE=	`Install` for an initial installation; `Modify` for an update.	*Install*

Table 8. WebSphere properties for silent installation

Property	Valid values	Example value
WAS_HOME=	Path to the WebSphere Application Server root directory on this node.	*/opt/IBM/WebSphere/AppServer*
WAS_PROFILE=	Path to the WebSphere Application Server profile directory on this node.	*/opt/IBM/WebSphere/AppServer/profiles/Appsrv01*
WAS_ADMIN_USER=	Administrative user name for WebSphere Application Server	*was_admin*
WAS_ADMIN_PASSWORD=	Password for the WebSphere Application Server administrative user	*waspass*
IS_CLUSTER	Indicates whether the installation is on a WebSphere cluster or a single server. Required only for deployment manager installations.	*true* or *false*
WAS_SERVER_NAME, **WAS_NODE_NAME**	The WebSphere server and node that will be used for the installation. Used when IS_CLUSTER=false and this is a Dmgr installation. Required when installing or modifying a deployment manager on a server scope.	*PS_Server* *node1*
WAS_CLUSTER_NAME, **WAS_PROXY_NAME**, **WAS_PROXY_NODE_NAME**	The WebSphere cluster name, proxy name and proxy node name. Used when IS_CLUSTER=true and this is a Dmgr installation. Required when installing or modifying a deployment manager on a cluster scope.	*PS_Cluster* *PS_Proxy* *node1*

Table 9. CSCF configuration properties for silent installation

Property	Valid values	Example value
CONFIGURE_CSCF	If true, the CSCF address is set in the WebSphere proxy configuration. If false, the CSCF address configuration is not set. Required when installing or modifying a deployment manager on a cluster scope.	*true* or *false*

Table 9. CSCF configuration properties for silent installation (continued)

Property	Valid values	Example value
CSCF_ADDRESS	Any valid CSCF address string. In a clustered configuration with CONFIGURE_CSCF=true, this is a semicolon-delimited list of hosts (IP addresses or host names) for which the proxy server will not remove the P-asserted identity headers that are added by the CSCF. Required when installing or modifying a deployment manager on a cluster scope. The use of a mask (*) is permitted. Do not include spaces in the input string.	*192.0.2.21;*.*.*.**

Table 10. TAI configuration properties for silent installation

Property	Valid values	Example value
CONFIGURE_TAI	1 to set the IMS Trust Association Interceptor (TAI) configuration in WebSphere Application Server. 0 to not set the configuration. Required only for deployment manager installations.	*0* or *1*
TAI_ALLOWED_SENDERS	Required when installing or modifying a deployment manager with the IMS TAI configured (CONFIGURE_TAI=1), this field assigns the value for the allowed senders configuration parameter. It is a comma-delimited list of hosts (IP addresses or host names) that the interceptor considers to be trusted. The use of a mask (**x** for the TAI) is permitted. Do not include spaces in the input string.	*x.example.com,x.x.x.x,2002:92A:8F7A:0:0:20:0:1*

Table 11. General database properties for silent installation

Property	Valid values	Example value
DATABASE_VENDOR	Database type for the Presence Server database–either DB2 or Oracle	*db2*
DB_JDBC_DIR	Path to the database client driver jars on the WebSphere Application Server host	*/opt/ibm/db2/V9.5/java*

Table 12. Presence Server database properties for silent installation

Property	Valid values	Example value
DB_HOST_NAME	Host name of the database server to be used for the store	*localhost*
DB_NAME	Name of the database to be used for the store	*PSDB*
DB_PORT	Port number on the database server to be used for the store	*50000*
DB_ADMIN_NAME	User with permissions for the database	*dbAdmin*
DB_ADMIN_PASSWORD	Password of the user with permissions for the database	*dbAdminPass*
UPDATE_PS_DB	1 indicates that the configuration table should be deleted. 0 indicates that the configuration table should remain the same as it was before the process. If the configuration is empty, the default configuration will be loaded.	*0* or *1*

Table 13. Content indirection database properties for silent installation

Property	Valid values	Example value
CI_DB_HOST_NAME	Host name of the database server to be used for the content indirection database	*localhost*
CI_DB_NAME	Name of the database to be used for the store	*PSCIDB*
CI_DB_PORT	Port of the database server to be used for the store	*50000*
CI_DB_ADMIN_NAME	User with permissions for the database	*dbAdmin*
CI_DB_ADMIN_PASSWORD	Password of the user with permissions for the database	*dbAdminPass*

Table 14. Usage record database properties for silent installation

Property	Valid values	Example value
UR_DB_HOST_NAME	Host name of the database server to be used for the Usage Records database	*localhost*
UR_DB_NAME	Name of the database to be used for the store	*PSURDB*
UR_DB_PORT	Port of the database server to be used for the store	*50000*
UR_DB_ADMIN_NAME	User with permissions for the database	*dbAdmin*
UR_DB_ADMIN_PASSWORD	Password of the user with permissions for the database	*dbAdminPass*

Table 14. Usage record database properties for silent installation (continued)

Property	Valid values	Example value
UR_UPDATE_PS_DB	1 indicates that the configuration table should be deleted. 0 indicates that the configuration table should remain the same as it was before the process. If the configuration is empty, the default configuration will be loaded.	*0* or *1*
CREATE_UR_TABLES	1 indicates that the USAGERECORDS table should be recreated. 0 indicates that the table should not be recreated. Used to prevent the deletion of existing usage records data.	*0* or *1*

Table 15. Data store database properties for silent installation

Property	Valid values	Example value
BUS_DATASTORE	1 indicates that the SIBus database will use Data Store; 0 indicates that the SIBus database will use File Store. Used if you want to preserve the Usage Records database tables after a modification.	*0* or *1*
BUS_DB_HOST_NAME	Host name of the database server to be used for the store	*localhost*
BUS_DB_NAME	Name of the database to be used for the store	*PSSIBUS*
BUS_DB_PORT	Port number on the database server to be used for the store	*50000*
BUS_DB_ADMIN_NAME	User with permissions for the database	*dbAdmin*
BUS_DB_ADMIN_PASSWORD	Password of the user with permissions for the database	*dbAdminPass*

3. Save the edited file.
4. Copy the edited files to each cluster member and deployment manager where you plan to install the Presence Server product.

What to do next

You are ready to run the installation response file using the **-i silent -f** *path* options. For details, see the topic *Installing silently*.

Running the silent installation

You can install IBM WebSphere Presence Server Component silently, using a response file to supply installation options without user interaction.

Before you begin

Configure the installation, by changing the values of the parameters in the response file. This process is detailed in the topic *Editing the response file*.

About this task

A silent installation uses the installation wizard to install the product in silent mode, without the graphical user interface. Instead of displaying a wizard interface, the silent installation causes the installation program to read all of your responses from a file that you provide. To specify non-default options during a silent installation, you must use the response file. To install silently, you must accept the license agreement in the agreement option.

1. Log on to the operating system.
2. Run the following command: `chmod 755 installer_path/setup.bin`, where *installer_path* is the fully qualified path name where the `setup.bin` file is located.
3. Change (`cd`) to the *installer_path* directory.
4. To use your custom response file, issue the proper command in the installer directory. For example:

   ```
   ./setup.bin -i silent -f path
   ```

   where *path* is the fully qualified path to the custom response file, for example *was_root*`/installableApps/presence/presenceServer_install.rsp`. The setup script performs the installation.
5. Optional: To review the installation logs, navigate to *was_root*`/logs/`.

   Note: *was_root* is the installation root directory for WebSphere Application Server Network Deployment. By default, this directory is:

   - AIX `/usr/IBM/WebSphere/AppServer`
   - Linux `/opt/IBM/WebSphere/AppServer`

What to do next

If this installation was performed on the deployment manager or on a federated node, perform the same operation on each node in the cluster.

Use the primary configuration file, `SystemConfiguration.xml`, if you would like to modify the default configuration before deploying the Presence Server application. This file is found in the directory *was_root*`/installableApps/presence/scripts/config`. (There is also a backup copy in *was_root*`/logs`.) For more information, refer to the *Configuring IBM WebSphere Presence Server Component* section of this information center.

Setting replication for the SIP container

For optimum performance, enable end-of-service replication for the SIP container on the server where the Presence Server application is installed.

Before you begin

Start WebSphere Application Server Network Deployment on the server where you installed the Presence Server product.

About this task

The `end.of.service.replication` option specifies that changes are buffered until the thread for a siplet is about to end. Setting this option is likely to improve system performance when Presence Server is operating.

For more information about these options, refer to the topic *SIP container custom properties* in the WebSphere Application Server 7.0 Information Center.

Note: For a complete, up-to-date list of recommended WebSphere customization settings, consult the document *Tuning IBM WebSphere Presence Server component*. You can download this document from the IBM Support Web site.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Enable end-of-service replication and disable immediate replication for the SIP container:
   a. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
   b. Click the name of the server on which you installed the Presence Server product.
   c. Under Container Settings, click **SIP Container Settings** → **SIP container**.
   d. Click **Custom properties**.
   e. Add the following custom properties:
      - end.of.service.replication = true
      - immediate.replication = false
   f. Click **Save** to save your changes.
3. Stop and restart the application server, nodes, and deployment manager.

Installing updates for the Presence Server product

You can apply updates to the Presence Server product, like interim fixes and fix packs, without disruption to normal processing when the product is deployed in a horizontal WebSphere cluster. This feature is known as a *rollout update*. In a vertical cluster, the same process is used to apply updates, but there will be an interval during which Presence Server is not available for use.

Before you begin

The EAR file for the Presence Server update you are installing, for example *was_root*/installableApps/presence/IBMWebSpherePresenceServer.ear, must exist either on the local computer or on the deployment manager node.

Refer to the Readme file, which is included with every update or fix, for additional prerequisites.

About this task

Note that continuous availability is possible only in clusters that have a horizontal scaling topology–that is, cluster members exist on multiple WebSphere nodes. In the case of a vertical configuration (where all cluster members exist on the same physical computer), you should follow the same steps to perform updates. However, the Presence Server application will be taken offline while you are applying the updates.

To install updates to the Presence Server product, follow these steps:

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
3. Select the Presence Server check box and click **Rollout Update**.

   Note: In WebSphere Application Server version 6.1.0.x, select the Presence Server check box and click **Update**.
4. Browse to the EAR file for the Presence Server update you are installing, for example *was_root*/installableApps/presence/IBMWebSpherePresenceServer.ear.
5. Optional: Make any desired changes to the default settings, clicking **Next** to navigate through the lists of options.
6. Click **Finish** to complete the update. When the update is complete, the following message displays: Application Presence Server installed successfully.
7. Click the **Rollout Update** link.
8. When all cluster members have completed the synchronization phase, click **Continue**.

Installing updates from the command line

About this task

As an alternative to using the Integrated Solutions Console, you can install updates using the `PresenceServerRollout.sh` script.

1. Log in to the deployment manager.
2. Change (`cd`) to the deployment manager profile directory.
3. Edit the file `properties/soap.client.props` and update the timeout property: `com.ibm.SOAP.requestTimeout=0`.
4. Copy the `PresenceServerRollout.sh` script from the installation medium to any directory on the deployment manager, for example `/usr` or `/opt`.
5. Change to the directory into which you copied `PresenceServerRollout.sh`.
6. Run the following command to ensure that the script will run properly:

   ```
   chmod 755 PresenceServerRollout.sh
   ```

7. Run the script:

   ```
   ./PresenceServerRollout.sh –dmgrProfile dmgr_profile_path -username user_name -password password
   ```

   where:

   dmgr_profile_path is the full path to the deployment manager profile directory.

   user_name represents your WebSphere Application Server administrator user ID.

   password represents the password associated with your *user_name*.

   ear_file_path is the path to the EAR file for the Presence Server update you are installing, for example *was_root*`/installableApps/presence/IBMWebSpherePresenceServer.ear`.

Adding a server to an existing Presence Server node

If you have a running system with a Presence Server node that was defined during installation, you can add a new server instance by performing the configuration using the WebSphere Integrated Solutions Console.

Before you begin

You must have an existing node that was created as part of a full installation of the Presence Server product, using the Presence Server installer.

Note: For a complete, up-to-date list of recommended WebSphere customization settings, consult the document *Tuning IBM WebSphere Presence Server component*. You can download this document from the IBM Support Web site.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.

b. Enter an administrator user ID and password.

c. Click **Log in**.

2. Create a new server in a cluster:

a. In the Integrated Solutions Console, click **Servers** → **Clusters** → **WebSphere application server clusters**.

Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Clusters**.

b. Select the check box associated with the name of the cluster.

c. Click **Start**.

d. Click the existing cluster name.

e. Under Additional Properties, click **Cluster members**.

f. Click **New**.

g. Specify the new server name.

h. From the drop-down list, select the node on which you want to create the new server.

i. Click **Add Member**.

j. Click **Next**.

k. Click **Finish**.

3. Add a new bus member:

a. Under Service integration, click **Buses**.

b. In the Buses list, click **PS_bus**.

c. Under Topology (on the Configuration tab), click **Bus members**.

d. Click **Add**.

e. Click **Server**.

f. From the drop-down list, select the server in the cluster you defined for the Presence Server application, for example WPS_server1. Then, click **Next**.

g. Click **File store** or **Data store** (according to your configuration), and click **Next**.

h. Click **Next**.

i. Click **Finish**.

j. Repeat the preceding steps, to add each server in the cluster.

k. Click **Save** to save changes to the master configuration.

l. Click **OK** when the node synchronization has completed.

4. Review the Session Initiation Protocol (SIP) and HTTP ports assigned to the new server in the cluster:

a. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.

b. Click the server name, for example *WPS_server1*.

c. Under Communications, click **Ports**.

d. Verify that SIP_DEFAULTHOST, SIP_DEFAULTHOST_SECURE, and WC_defaulthost appear in the list. Remember these port numbers, you will need them later.

e. Also verify these values for the other new servers added to the cluster.

5. Add definitions for virtual hosts:
   a. On the navigation panel, click **Environment** → **Virtual Hosts**.
   b. Click **default_host**.
   c. Under Additional Properties, click **Host Aliases** to display a list of port numbers.
   d. Verify that the port numbers collected in step 4 on page 59 are specified in the list of ports.
   e. Click **New** to create the missing port.
   f. Type the port number in the **Port** field.
   g. Click **OK**.
   h. Add additional ports until all port numbers are listed.
6. Create a thread pool for the SIP container:
   a. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
   b. Click *new_server_name*.
   c. Under Additional Properties, click **Thread pools**.
   d. Click **New**, to create a thread pool for the SIP container:
      - Name: SipContainer
      - Minimum Size: 60
      - Maximum Size: 60
      - Thread inactivity timeout: 5000 (default value)

      Note: You can modify these values according to your needs. The *Presence Server Performance Tuning Guide*, available as a technote on the IBM Support Web site, offers recommended values.
   e. Click **OK** and then **Save**, to save the values.
7. Update the default thread pool:
   a. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
   b. Click *new_server_name*.
   c. Under Additional Properties, click **Thread pools**.
   d. Click **Default**, and update the following values:
      - Minimum Size: 60
      - Maximum Size: 60
   e. Click **OK** and then **Save**, to save the values.
8. Set the SIP container settings:
   a. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.

b. Click *new_server_name*.
c. Under Container Settings, click **SIP Container Settings → SIP container**.
d. Set the SIP container's general properties as the following:
   - Maximum application sessions: 200000
   - Maximum messages per averaging: 200000
   - Maximum dispatch queue size: 5000
   - Thread pool: SipContainer (created in step 7 on page 60)
e. Click **Apply**.
f. Click **Custom properties**.
g. Add the following custom properties:
   - app.composition.enabled = false
   - com.ibm.ws.sip.security.enable.digest.tai = false
   - end.of.service.replication = true
   - immediate.replication = false
h. Click **Save**, to save your changes.

9. Set the EJB timer threads settings:
   a. In the navigation pane, click **Servers → Server Types → WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers → Application servers**.
   b. Click *new_server_name*.
   c. Under Container Settings, click **EJB Container Settings → EJB timer service settings**
   d. Change the number of timer threads to 20.
   e. Click **OK** and then **Save**, to save the values.
10. Set the ORB SERVICES settings:
   a. In the navigation pane, click **Servers → Server Types → WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers → Application servers**.
   b. Click *new_server_name*.
   c. Under Container Settings, click **Container Services**.
   d. Click **ORB Service** and select **Pass by reference**.
   e. Click **Apply** and then **Save**, to save the values.
11. Set the JVM settings:
   a. In the navigation pane, click **Servers → Server Types → WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers → Application servers**.
   b. Click *new_server_name*.
   c. Under Server Infrastructure, click **Java and Process Management → Process Definition**.
   d. Under Additional Properties, click **Java Virtual Machine**, and set the following properties:
      - Initial Heap Size = 1600

- Maximum Heap Size = 1600
- Generic JVM arguments = -Xgcpolicy:gencon -Xgc:scvnoAdaptiveTenure,scvTenureAge=3,stdGlobalCompactToSatisfyAllocate -Xconcurrentlevel1 -Xmn400m -Xmo1200m -Xtgc:parallel -Xdump:heap:events=user,request=exclusive+prepwalk+compact

  Note: You can copy and paste the generic JVM arguments above into the **Generic JVM arguments** field.

e. Click **Apply** and then **Save**, to save the values.

12. Stop and restart the application server, nodes, and deployment manager.

Uninstalling IBM WebSphere Presence Server Component

Follow these procedures to uninstall IBM WebSphere Presence Server Component.

Uninstalling Presence Server using the interactive uninstaller

You can uninstall the IBM WebSphere Presence Server Component by using the interactive uninstaller, an uninstall wizard.

About this task

To uninstall Presence Server follow these steps:

1. Change (`cd`) to the *`was_root`*`/Uninstall_PresenceServer` directory, where *was_root* is the installation root directory for WebSphere Application Server Network Deployment.
2. Set the proper file permissions on the uninstaller by entering the following from the command prompt:

   ```
   chmod 755 uninstaller
   ```

3. Launch the interactive installer. Run the following command: `./uninstaller`
4. In the Introduction pane, click **Next**.
5. In the Uninstall Details pane, perform one of the following choices:
   - For a deployment manager node or a stand-alone server, click **Next**.
   - For a clustered node that is not a deployment manager, click **Uninstall** and skip Step 6.
6. Type the administrative user's name and password, and click **Uninstall**. For example:

   WebSphere Administrator Name: `was_admin`

   WebSphere Administrator Password: `waspass`

Uninstalling Presence Server silently

As an alternative to using the wizard, the IBM WebSphere Presence Server Component can be uninstalled silently without user interaction, by reading user responses from a response file.

A silent uninstall performs the removal of the product in silent mode, without the interactive uninstaller. Instead of displaying the wizard interface, the silent uninstall allows the uninstall program to read all of your responses from a file that you provide. To specify non-default options during a silent uninstall, customize the response file.

Editing the uninstall response file

Before invoking the Presence Server uninstaller in silent mode, you need to customize the response file that will be used to provide user responses to the uninstaller.

About this task

Use the response file, `presenceServer_uninstall.rsp`, to supply values to the uninstaller. Running in silent mode, the uninstaller does not display interactive dialogs. Instead, it reads values from the response file.

Customize the response file precisely so that the installation program can read the option values accurately. For example, always enclose values in double quotation marks. When the installer encounters an incorrect parameter, it stops and writes an explanation of the problem to the installer log file: `presenceServerInstall.log`.

Note: The response file is required only if you are uninstalling from a deployment manager node or a stand-alone server. If you are uninstalling from a clustered node that is not a deployment manager, you do not need to edit the file.

Perform the following steps to edit the response file.

1. Navigate to *was_root*`/installableApps/presence`.

   Note: *was_root* is the installation root directory for WebSphere Application Server Network Deployment. By default, this directory is:

   - AIX `/usr/IBM/WebSphere/AppServer`
   - Linux `/opt/IBM/WebSphere/AppServer`

2. Open the uninstall response file, `presenceServer_uninstall.rsp`, in a text editor.

   Note: As an alternative to the default uninstall response file, you can build a response file using the template found on the Presence Server CD in the `/response_files` directory: `presenceServer_uninstall_template.rsp`. To use this file for silent uninstallation, copy it to each cluster member and each deployment manager, then modify it .

3. Edit the response file, modifying the following values as needed. The following table lists the response file parameters.

Table 16. Response file WebSphere variable properties

property	valid values	example value
WAS_ADMIN_USER=	Administrative user name for WebSphere Application Server	*was_admin*
WAS_ADMIN_PASSWORD=	Password for the WebSphere Application Server administrative user	*waspass*

4. Save the edited file.
5. Copy the edited files to each cluster member and deployment manager where you plan to uninstall the Presence Server product.

What to do next

You are now ready to run the uninstall script. The script will use the edited response file to get the settings it needs. See the topic *Running the silent uninstall* for details.

Running the silent uninstall

After you edit the response file, you can uninstall the Presence Server product silently, without user interaction.

Before you begin

Before beginning the silent uninstall, you need to configure the parameters values in the response file, `presenceServer_uninstall.rsp`. Refer to the topic *Editing the uninstall response file* for details.

About this task

Follow these steps to uninstall Presence Server in silent mode:

1. Log on to the operating system.
2. Change (`cd`) to the *`was_root`*`/Uninstall_PresenceServer` directory, where *was_root* is the installation root directory for WebSphere Application Server Network Deployment.
3. Set the proper file permissions on the uninstaller by entering the following from the command prompt:

   ```
   chmod 755 uninstaller
   ```

4. To uninstall the Presence Server product using your custom response file, issue the following command. For example:

   ```
   ./uninstaller -i silent -f path
   ```

   where *path* is the fully qualified path name of your response file.
5. Optional: To examine the logs, view the file *`was_root`*`/logs/presenceServerUninstall.log log`.

What to do next

If this uninstallation was performed on the deployment manager or on a federated node, perform the same operation on each node in the cluster.

Dropping the databases

After you have uninstalled the Presence Server program code and other components, you can drop the databases.

Removing the DB2 databases and database tables

The following topics describe how to clean and remove DB2 database tables, and how to remove the databases, when uninstalling IBM WebSphere Presence Server Component.

About this task

As part of the installation of the Presence Server product, you created some or all of the following databases:

- The Presence Server application database: *PSDB*
- The Presence Server content indirection database: *PSCIDB*

- The Presence Server usage record database: *PSURDB*
- The Presence Server data store database: *PSSIBUS*

Follow these steps to clean and remove the database tables and remove the databases:

1. Log in to the DB2 server as a database administrator.
2. Create a directory that has write and execute permission, for example `DB_dir`.
3. Copy the installation `.tar` file, `IBMPresenceServerDbPackage_7.0.tar`, from the installation medium to the new directory.
4. Switch to the new directory.
5. Unpack the installation `.tar` file using the following command:

   ```
   tar -xvf IBMPresenceServerDbPackage_7.0.tar
   ```
6. Switch to the following directory: *was_root*`/installableApps/presence/scripts/dbScripts/presence`.
7. Verify that the following files exist in `/presence`:
   - `ConfigDB2.sh`
   - `RunConfigDB2.sh`
   - `RunConfigContentIndirectionDB2.sh`
   - `DropDB2Tables.ddl`
   - `DropContentIndirectionDB2Tables.ddl`
8. Run the following commands to ensure that the uninstall scripts will run properly:

   ```
   chmod 755 ConfigDB2.sh
   chmod 755 RunConfigDB2.sh
   chmod 755 RunConfigContentIndirectionDB2.sh
   ```
9. Edit the Presence Server database preparation script, `RunConfigDB2.sh`, for your environment:
   a. Open `RunConfigDB2.sh` using a text editor.
   b. Locate the line that begins with the following text:

      ```
      #Update command line parameters here.
      ```
   c. Update values for each parameter listed:

Parameter	Description	Default value
DBSERVER	Fully qualified name of the database server Linux Should match the Linux `hostname` command value if you are running the script directly on the database server	<hostname>
DBPORT	Number for the listening port on the database server	50000
DBNAME	Name of the database	PSDB
DBALIAS	Alias by which the database is known	PSDB
DBLOCALE	Territory code that identifies your DB2locale, for example US or JP	US

Parameter	Description	Default value
DBINSTANCE	Name of the database instance	db2inst
DBINSTANCPW	Password for the database instance	<pw>
DBUSER	User ID for the authorized user (the ID with which you are logged in)	db2inst
DBUSERPW	Password for the authorized user	<pw>
DBDIR	Directory in which the database files are to be created (should be a directory with write permission)	/home/db2inst
DBCREATE	Boolean value specifying whether the database should be recreated	FALSE

For example:

```
DBSERVER=dbserver.example.com
DBPORT=50000
DBNAME=PSDB
DBALIAS=PSDB
DBLOCALE=US
DBINSTANCE=db2inst
DBINSTANCPW=inst_pw
DBUSER=db2inst
DBUSERPW=user_pw
DBDIR=/home/db2inst
DBCREATE=FALSE
```

d. Locate the section of the script titled `Create Presence Server tables`, at the end of the script, and replace the double-quote ("") with the name of the ddl file: `DropDB2Tables.ddl`.

e. Save your changes.

f. Clean Presence Server database tables by running the `RunConfigDB2.sh` script.

10. Optionally, remove the Presence Server database by running the following commands:

```
db2 "drop database" dbname
db2 "uncatalog database" aliasname
db2 "terminate"
```

where:

dbname is the database name.

aliasname is the alias by which the database is known.

11. Edit the content indirection database preparation script, `RunConfigContentIndirectionDB2.sh script`, for your environment:

a. Open `RunConfigContentIndirectionDB2.sh` using a text editor.

b. Locate the line that begins with the following text:

```
#Update command line parameters here.
```

c. Update values as you did in step 9 on page 65.

d. Locate the section of the script titled `Create Presence Server tables`, at the end of the script, and replace the double-quote ("") with the name of the ddl file: `DropContentIndirectionDB2Tables.ddl`.

e. Save your changes.

f. Clean content indirection database tables by running the `RunConfigContentIndirectionDB2.sh` script.

12. Optionally, remove the content indirection database:

    a. Connect to the content indirection database.

    b. Remove the database by running the following commands:

    ```
    db2 "drop database" dbname
    db2 "uncatalog database" aliasname
    db2 "terminate"
    ```

    where:

    dbname is the name of the content indirection database, for example PSCIDB.

    aliasname is the alias by which the content indirection database is known.

13. Remove the usage records database tables, tablespace, and buffer pool. A DDL file, `DropUsageDbDb2`, is provided for this task. The DDL file is found in the directory *was_root*`/installableApps/presence/scripts/dbScripts/usageRecords`.

    a. Connect to the usage records database by running the following command:

    ```
    db2 connect to database_alias user database_administrator_ID
    using database_administrator_PW
    ```

    Where:

    database_alias represents the database alias

    database_administrator_ID represents the administrator user ID

    database_administrator_PW represents the administrator password

    b. Run the following commands:

    ```
    db2 -tvf DropUsageDbDb2.ddl
    db2 connect reset
    db2 "terminate"
    ```

14. Remove the usage records database:

    a. Connect to the usage records database.

    b. Remove the database by running the following commands:

    ```
    db2 "drop database" dbname
    db2 "uncatalog database" aliasname
    db2 "terminate"
    ```

    where:

    dbname is the name of the usage records database, for example PSURDB.

    aliasname is the alias by which the usage records database is known.

15. Optional: Remove the SIBus database:

    a. Connect to the SIBus database.

    b. Remove the database by running the following commands:

    ```
    db2 "drop database" dbname
    db2 "uncatalog database" aliasname
    db2 "terminate"
    ```

    where:

    dbname is the name of the SIBus database, for example PSSIBUS.

aliasname is the alias by which the SIBus database is known.

16. If you did not delete the SIBus database in step 15 on page 67, delete the database schemas on the deployment manager.

    Note: Failure to delete the SIBus schemas could result in the SIBus not starting properly after an upgrade or a reinstallation.

    a. Stop the cluster.
    b. Log in to the deployment manager.
    c. Change (cd) to the following directory: *was_root*/installableApps/presence/scripts/dbScripts/SIBus.
    d. Verify that the file dropDB2SIBusSchemas.sh exists in the /SIBus directory.
    e. Run the following command to ensure that the script will run properly:

       ```
       chmod 755 dropDB2SIBusSchemas.sh
       ```

    f. Run the script to delete the SIBus schemas:

       ```
       ./dropDB2SIBusSchemas.sh java_home jar_name jdbc_jars_dir db_user db_password db_server db_na
       ```

       where:

       java_home is the Java home directory, for example *was_root*/java/bin.

       jar_name is the full path to the database JAR file, for example /tmp/DataBase/UpdateConfiguration.jar. (The JAR file is located inside the Presence Server installation .tar file, IBMPresenceServerDbPackage_7.0.tar.)

       jdbc_jars_dir is the directory that contains JDBC driver JAR files for the database client, for example /opt/db2_jars.

       db_user is the database user, for example dbAdmin.

       db_password is the password for the database user.

       db_server is the name of the database server, for example serv1 or localhost.

       db_name is the database name, for example PSSIBUS.

       db_port is the database port, for example 50000.

       db_schema_file is the full path to the file that contains the SIBus schemas list. By default this file is *was_root*/logs/presenceServerInstall.log.bus_schemas.

Removing the Oracle database tables

The following topics describe how to clean and remove Oracle database tables when uninstalling IBM WebSphere Presence Server Component.

About this task

As part of the installation of the Presence Server product, you created some or all of the following databases:

- The Presence Server application database: *PSDB*
- The Presence Server content indirection database: *PSCIDB*
- The Presence Server usage record database: *PSURDB*
- The Presence Server data store database: *PSSIBUS*

Follow these steps to clean and remove the database tables:

1. Log in to the Oracle server as a database administrator.
2. Create a directory that has write and execute permission, for example DB_dir.

3. Copy the installation .tar file, IBMPresenceServerDbPackage_7.0.tar, from the installation medium to the new directory.
4. Switch to the new directory.
5. Unpack the installation .tar file using the following command:

   ```
   tar -xvf IBMPresenceServerDbPackage_7.0.tar
   ```
6. Switch to the following directory: *was_root*/installableApps/presence/scripts/dbScripts/presence.
7. Verify that the following files exist in /presence:
   - ConfigOracle.sh
   - RunConfigOracle.sh
   - RunConfigContentIndirectionOracle.sh
   - DropOracleTables.ddl
   - DropContentIndirectionOracleTables.ddl
8. Run the following commands to ensure that the uninstall scripts will run properly:

   ```
   chmod 755 ConfigOracle.sh
   chmod 755 RunConfigOracle.sh
   chmod 755 RunConfigContentIndirectionOracle.sh
   ```
9. Edit the Presence Server database preparation script, RunConfigOracle.sh, for your environment:
   a. Open RunConfigOracle.sh using a text editor.
   b. Locate the line that begins with the following text:

      ```
      #Update command line parameters here.
      ```
   c. Update values for each parameter listed:

Parameter	Description	Default value
DBNAME	Name of the database	PSDB
DBUSER	User ID for the authorized user (the ID with which you are logged in)	<user name>
DBUSERPW	Password for the authorized user	<pw>

      For example:

      ```
      DBNAME=PSDB
      DBUSER=user_ID
      DBUSERPW=user_pw
      ```
   d. Locate the section of the script titled Create Presence Server tables, at the end of the script, and replace the name of the ddl file with DropOracleTables.ddl.
   e. Save your changes.
   f. Run the RunConfigOracle.sh script.
10. Edit the content indirection database preparation script, RunConfigContentIndirectionOracle.sh, for your environment:
    a. Open RunConfigContentIndirectionOracle.sh using a text editor.
    b. Locate the line that begins with the following text:

       ```
       #Update command line parameters here.
       ```
    c. Update values as you did in step 9.

d. Locate the section of the script titled `Create Presence Server tables`, at the end of the script, and replace the name of the ddl file with `DropContentIndirectionOracleTables.ddl`.
e. Save your changes.
f. Run the `RunConfigContentIndirectionOracle.sh` script.

11. Remove the usage records database tables:
    a. Connect to the usage records database by running the following command:
       `sqlplus $dbuser/$dbuserPW@$dbname`
       where:
       dbsser is the user ID for the authorized user.
       dbuserPW is the password for the authorized user.
       dbname is the name of the usage records database, for example PSURDB.
    b. Remove the database tables by running the following command:
       `DROP TABLE USAGERECORDS;`
12. Optional: Remove the SIBus database tables:
    a. Connect to the SIBus database by running the following command:
       `sqlplus $dbuser/$dbuserPW@$dbname`
       where:
       dbsser is the user ID for the authorized user.
       dbuserPW is the password for the authorized user.
       dbname is the name of the SIBus database, for example PSSIBUS.
    b. Remove the database tables by running the following command for each table:
       `DROP TABLE table_name;`
13. If you did not delete the SIBus database in step 12, delete the database schemas on the deployment manager.

    Note: Failure to delete the SIBus schemas could result in the SIBus not starting properly after an upgrade or a reinstallation.
    a. Stop the cluster.
    b. Log in to the deployment manager.
    c. Change (`cd`) to the following directory: *was_root*`/installableApps/presence/scripts/dbScripts/SIBus`.
    d. Verify that the file `dropOracleSIBusSchemas.sh` exists in the `/SIBus` directory.
    e. Run the following command to ensure that the script will run properly:
       `chmod 755 dropOracleSIBusSchemas.sh`
    f. Run the script to delete the SIBus schemas:
       `./dropOracleSIBusSchemas.sh java_home jar_name jdbc_jars_dir db_user db_password db_server db_name db_port db_schema_file`

       where:
       java_home is the Java home directory, for example *was_root*`/java/bin`.
       jar_name is the full path to the database JAR file, for example `/tmp/DataBase/UpdateConfiguration.jar`. (The JAR file is located inside the Presence Server installation `.tar` file, `IBMPresenceServerDbPackage_7.0.tar`.)

jdbc_jars_dir is the directory that contains JDBC driver JAR files for the database client, for example `/opt/oracle_jars`

db_user is the database user, for example root.

db_password is the password for the database user.

db_server is the fully qualified name of the database server host, for example `dbserver.example.com`.

db_name is the database name, for example PSSIBUS.

db_port is the database port, for example 1521.

db_schema_file is the full path to the file that contains the SIBus schemas list. By default this file is *was_root*`/logs/presenceServerInstall.log.bus_schemas`.

Chapter 4. Configuring IBM WebSphere Presence Server Component

You can modify the default configuration for the IBM WebSphere Presence Server Component and then make updates as the network environment changes.

Most configuration is managed using the XML configuration file, `SystemConfiguration.xml`. The data in this file is written to the Presence Server database and thus is shared among all instances of IBM WebSphere Presence Server in the cluster.

To make configuration changes after the IBM WebSphere Presence Server Component is installed, update the XML configuration file and then use the `UpdateConfiguration.jar` file to update the database. To implement the configuration changes, restart all of the servers.

The following topics contain detailed instructions for updating the configuration for various aspects of your Presence Server deployment.

Configuring for integration with external sources

You can configure Presence Server to gather presence information from external sources including the Serving-Call/Session Control Function (S-CSCF).

Presence Server provides a mechanism for gathering presence information from SIP external sources and registration information from S-CSCF.

The external SIP server, also referred to as a SIP provider or SIP external source, is addressed only when a subscriber is seeking information about a presentity. As a result, you can configure Presence Server so that the presence information for the presentity is collected only when it is required.

The information received from external sources is stored and handled using the same mechanism used for incoming PUBLISH requests. As a result, this additional information is included in the composed full document for the presentity and can be accessed by all subscribers for the presentity.

A commonly used example of a SIP external source is the Serving-Call/Session Control Function (S-CSCF).

Configuring for SIP external sources

Follow these steps to configure Presence Server to interact with SIP external sources.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`

- UpdateConfiguration.jar

Before you can make changes to your configuration, the file UpdateConfiguration.jar must be in the class path.

Note that the file UpdateConfiguration.jar must be in the class path.

Before you can make configuration changes, you will need to know the external source's SIP address–starting with either sip or sips.

Note: For sips addresses, Transport Layer Security (TLS) is the only supported protocol.

About this task

This value will appear in the P-Asserted-Id: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. The SIP external source examines the p-asserted-identity header to verify that it relates to an authorized user.

To configure the SIP external source, enable the External Sources mechanism and provide the authenticated identity of the Presence Server. You can configure the SIP external source by completing the following steps:

1. Open SystemConfiguration.xml with a text editor.
2. Within the <externalSources> element, create a separate externalSource tag for each external source you want to define. Each externalSource tag defines a SIP external source and sets the protocols for communicating with it.

   You can set the following attributes for each external source. Default values, where applicable, are shown.

   enable="false"
   true to enable integration with this SIP external source; false to disable it

   sipAddress=""
   SIP address of the external source

   fromURI=""
   A valid SIP URI that will appear in the From: header for outgoing SUBSCRIBE requests sent to the external source.

   assertedIdentity=""
   URI, in scheme:identity format, of the requester authorized to get information from the SIP external source. The SIP external source examines the p-asserted-identity header to verify that it relates to an authorized user.

   For the assertedIdentity attribute, the display name is optional. If you choose to define a display name, you must use the HTML code for symbols for the value of the attribute. For example, use

   ```
   <assertedIdentity="&quot;Super Admin&quot; &lt;glm:GLMSuperAdmin&gt;"/>
   ```

   instead of

   ```
   <assertedIdentity=""Super Admin" <glm:GLMSuperAdmin>" />
   ```

   subscribeExpiration="60"
   Time, in minutes, to specify in the Expire header when Presence Server

sends SUBSCRIBE requests to the external source. The subscription will be refreshed as long as there are client subscriptions on the server.

retryInterval="5"
Time, in minutes, to wait before retrying a request that fails. This applies specifically to outgoing SUBSCRIBE requests sent to the external source. The valid range is 1-1440.

triggerEventPackage=""
The event package with which the external source is associated

sourceEventPackage=""
The event package used for outgoing subscriptions against the SIP external source

acceptHeader=""
The values used for the Accept header in outgoing subscriptions in comma-separated format. for example `accept1,accept2`.

If you leave this value blank, Presence Server sends outgoing subscription requests without an Accept header.

specifyAssertedIdentityOnAllRequests="true"
If set to `true`, a P-Asserted-Identity must be specified in all outgoing subsequent subscribe requests. If set to `false`, a P-Asserted-Identity is required only in the first outgoing SUBSCRIBE request but is not required in subsequent SUBSCRIBE requests.

Here is an example of an `externalSource` tag:

```
<externalSource enable="true"
sipAddress="sip:server.example.com:7010;transport=UDP"
                fromURI="sip:Admin1@example.com" assertedIdentity="
&quot;Super Admin1&quot;&lt;sip:ps@example.com&gt;"
subscribeExpiration="20" retryInterval="10"
triggerEventPackage="presence" sourceEventPackage="presence"
                acceptHeader=""
specifyAssertedIdentityOnAllRequests="true"/>
```

3. On the `waitForExternalInformation` tag, modify the `maxInterval` attribute to specify how long to wait while collecting information from IBM XDMS and other external sources before notifying the client.

   `maxInterval` specifies the time, in seconds, to wait while collecting information from IBM XDMS and other external sources before sending a notification to the client. For group subscriptions, the time is measured after the group content is received from IBM XDMS.

   Valid values are 0-60. The default value is 15.

   Note: Avoid setting the interval too small, to avoid having notifications sent before all information has been collected. It is also recommended that you select a value of 30 or below because a subscriber may invalidate the subscription if a notification is not returned within 30 seconds.

   For example:

   ```
   <waitForExternalInformation maxInterval="15" />
   ```

4. Save and close the file.
5. Open `ConfigurationParams.txt` with a text editor.
6. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

username = *database_administrator_user_name*

password = *database_administrator_password*

DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*

7. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX
   ```
   /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

   Linux
   ```
   /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

8. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.

   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring Presence Server to interact with S-CSCF sources

Follow these steps to configure Presence Server to interact with external sources for the Serving-Call/Session Control Function (S-CSCF).

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

To configure Presence Server to work with S-CSCF, you will need the following information:

- SIP address of the S-CSCF server
- P-asserted-identity of the requester authorized to get information from the S-CSCF server
- URI for the authorized S-CSCF user

About this task

This value will appear in the P-Asserted-Id: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. The S-CSCF examines p-asserted-identity header verifying it relates to an authorized user.

To configure integration with S-CSCF, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the attributes on the `S-CSCF` tag for your preferred configuration. This tag identifies the S-CSCF server and establishes parameters for communicating with it.

   You can set the following attributes. By default, `enable` is set to `false` and the other attributes are blank.

   enable=*"false"*
   : `true` to enable integration with S-CSCF; `false` to disable it

   sipAddress=""
   : Address of the S-CSCF server to which SUBSCRIBE requests are sent

   fromURI=""
   : A valid SIP URI that will appear in the From: header for outgoing SUBSCRIBE requests sent to the S-CSCF.

   assertedIdentity=""
   : URI, in scheme:identity format, of the requester authorized to get information from the S-CSCF server. The S-CSCF examines p-asserted-identity header verify it relates to an authorized user

     For the `assertedIdentity` attribute, the display name is optional. If you choose to define a display name, you must use the HTML code for symbols for the value of the attribute. For example, use

     ```
     <assertedIdentity="&quot;Super Admin&quot; <glm:GLMSuperAdmin>"/>
     ```

     instead of

     ```
     <assertedIdentity=""Super Admin" <glm:GLMSuperAdmin>" />
     ```

   subscribeExpiration="63"
   : Time, in minutes, to specify in the Expire header when Presence Server sends SUBSCRIBE requests to the S-CSCF. The subscription will be refreshed as long as there are client subscriptions on the server.

   retryInterval="5"
   : Time, in minutes, to wait before retrying a request that fails. This applies specifically to outgoing SUBSCRIBE requests sent to the S-CSCF. The valid range is 1-1440.

specifyAssertedIdentityOnAllRequests="true"
If set to `true`, a P-Asserted-Identity must be specified in all outgoing subsequent subscribe requests. If set to `false`, a P-Asserted-Identity is required only in the first outgoing SUBSCRIBE request but is not required in subsequent SUBSCRIBE requests.

Here is an example of an `S-CSCF` tag:

```
<S-CSCF enable="true"
sipAddress="sip:server.example.com:7010;transport=UDP"
                fromURI="sip:Admin1@example.com" assertedIdentity="
&quot;Super Admin1&quot;&lt;sip:ps@example.com&gt;"
subscribeExpiration="63" retryInterval="5"
specifyAssertedIdentityOnAllRequests="true"/>
```

3. Save and close the file.
4. Open `ConfigurationParams.txt` with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

   Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*
6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX
   ```
   /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

   Linux
   ```
   /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```
7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.

c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring for integration with IBM XDMS

Use the configuration file to define the way in which Presence Server uses the group list server, for example the functionality provided in IBM WebSphere XML Document Management Server Component (IBM XDMS), to retrieve public groups content.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

To configure Presence Server to communicate with an IBM XDMS server, you will need the following information:

- SIP address, port and transport protocol for the IBM XDMS server. The transport protocol is optional; if left blank it defaults to UDP.
- P-asserted-identity, which should include the SuperAdmin user defined in IBM XDMS
- IBM XDMS superuser ID and password with authority to query the user groups that you are trying to access (required only if the Aggregation Proxy is deployed with IBM XDMS)
- HTTP URL of the IBM XDMS server from which Presence Server should get data
- (For storing group lists) A valid SIP URI that will appear in the From: header for outgoing SUBSCRIBE requests sent to the server
- (For enabling authorization lists) Paths on the server for white-list and black-list definitions

Finally, before you can make these configuration changes, the Presence Server database must be running.

About this task

To configure Presence Server so that it can use IBM XDMS to get information about group lists, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the attributes on the `groupListServer` tag to identify the IBM XDMS server and user.

   You can set the following attributes. Default values are shown.

enable="false"
`true` to enable integration with an IBM XDMS server; `false` to disable it

sipAddress=""
Address and port of the IBM XDMS instance to which SUBSCRIBE requests are sent. The transport protocol for incoming requests can be specified here. If the transport protocol is omitted, then the XDMS listening port is configured in UDP protocol by default.

assertedIdentity=""
URI, in scheme:identity format, of the requester authorized to get XCAP documents from the IBM XDMS shared list server. This value is required.

This value will appear in the P-Asserted-Id: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. The identity must have IBM XDMS administrator authority to access user groups.

For the `assertedIdentity` attribute, the display name is optional. If you choose to define a display name, you must use the HTML code for symbols for the value of the attribute. For example, use

```
<assertedIdentity="&quot;Super Admin&quot; &lt;glm:GLMSuperAdmin&gt;"/>
```

instead of

```
<assertedIdentity=""Super Admin" <glm:GLMSuperAdmin>" />
```

user=""
A user ID with administrator-level privileges, for connecting to the Aggregation Proxy. This value is optional.

password=""
Password of the administrator-level user, for connecting to the Aggregation Proxy. Required only when a user is specified.

fromURI=""
A valid SIP URI that will appear in the From: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. This value is required.

retryInterval="5"
Time, in minutes, to wait before retrying a request that fails. This applies to outgoing SUBSCRIBE requests sent to an external source–for example, the IBM XDMS shared list server. Presence Server sends the requests to subscribe to the group content and to receive notification when the group content is modified. The valid range is 1-1440.

subscribeExpiration="63"
Time, in minutes, to specify in the Expire header when Presence Server sends SUBSCRIBE requests to the IBM XDMS shared list server. The subscription will be refreshed as long as there are client subscriptions on the server.

xcapRoot=""
HTTP URL of the IBM XDMS server from which Presence Server should get data using `xcap_get` requests. This attribute provides a way for Presence Server to connect directly with the IBM XDMS server rather than connecting through the Aggregation Proxy.

When xcapRoot is specified, Presence Server uses this value as the URI for performing the xcap_get operation–bypassing the Aggregation Proxy. In this case, Presence Server uses the asserted identity to perform the operation.

When xcapRoot is not specified, Presence Server performs the xcap_get operation using the xcap root URI that is specified in the NOTIFY request received from IBM XDMS. In this case, Presence Server uses the specified user ID and password to perform the operation.

You must specify an xcapRoot when enableXcapEvent is true.

enableXcapEvent= ″ ″
If set to true, Presence Server will use the xcap-diff event package for communication with the IBM XDMS server. If set to false (the default), Presence Server will use the ua-profile event package.

specifyAssertedIdentityOnAllRequests=″true″
If set to true, a P-Asserted-Identity must be specified in all outgoing subsequent subscribe requests. If set to false, a P-Asserted-Identity is required only in the first outgoing SUBSCRIBE request but is not required in subsequent SUBSCRIBE requests.

Here is an example of a groupListServer tag:

```
<groupListServer enable="true"
sipAddress="sip:server.example.com:7010;transport=UDP"
                  assertedIdentity="&quot;Super Admin1&quot;
&lt;sip:ps@example.com&gt;" user="glsuser" password="authpw"
fromURI="sip:Admin1@example.com"                  retryInterval="5"
subscribeExpiration="63" xcapRoot="http://
xdmshostname.example.com:9086/services"
enableXcapEvent="true" specifyAssertedIdentityOnAllRequests="true"/>
```

3. Save and close the file.
4. Open ConfigurationParams.txt with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for SystemConfiguration.xml)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

   Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*
6. Run the java command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for ConfigurationParams.txt)

Important: JDBC drivers must be separated by a colon.
For example:

AIX
```
/usr/IBM/WebSphere/AppServer/java/bin/java -classpath
UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
```

Linux
```
/opt/IBM/WebSphere/AppServer/java/bin/java -classpath
UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
```

7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Results

Note: `xcap_get` requests for external lists group lists uses user and password, if available. If there is no configured user, asserted identity is used `assertedIdentity`.

Configuring authorization and authentication

Authorization is the process of verifying that a presentity has the necessary authority to perform certain operations. Authentication is the process of checking incoming requests to ensure their validity. Use the Presence Server configuration file to enable or disable authorization and authentication, and to configure other security-related settings.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

About this task

In Presence Server, the configuration file is used to enable and disable authorization and to establish paths for allowing and disallowing access by white-list and black-list users. Presence Server also supports authorization rules, such as those in IBM WebSphere XML Document Management Server Component (IBM XDMS). These rules enable presentities to specify which users are allowed to watch their presence information.

Authorization is disabled by default. If you plan to use an authorization application with Presence Server, you must enable authorization (`externalAuthorization` tag) *and* you must disable white lists and black lists (`authorizationLists` tag).

To enable or disable authorization and authentication, and to configure other security-related settings, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the `externalAuthorization` tag to enable the use of external authorization applications.

   Set `enable="`*`true`*`"` to enable the authorization API. Subscriptions are authorized only when the authorization service provides authorization.

   Set `enable="`*`false`*`"` Address and port of the group list server to which SUBSCRIBE requests are sent. Transport protocol for incoming requests can be specified here. If transport protocol is omitted, then XDMS listening port is configured in UDP protocol by default. For example:

   ```
   <externalAuthorization enable="true"/>
   ```
3. Modify the attributes on the `authorizationLists` tag to define IBM XDMS paths for accessing a white list and a black list. Requests coming from presentities in the white list are always authorized, regardless of presence authorization rules. Requests coming from presentities in the black list are always rejected.

   You can set the following attributes. Default values are shown.

   enable=*"false"*
   : `true` to enable the use of white lists, black lists, or both; `false` to disable

   XDMSSipAddress=*""*
   : Address of the IBM XDMS server to which subscription requests are sent

   fromUri
   : A valid SIP URI that will appear in the From: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. This value is required.

   assertedIdentity=*""*
   : URI, in scheme:identity format, of the requester authorized to get XCAP documents

     This value will appear in the P-Asserted-Id: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. The identity must have IBM XDMS administrator authority to access user groups.

     For the `assertedIdentity` attribute, the display name is optional. If you choose to define a display name, you must use the HTML code for symbols for the value of the attribute. For example, use

     ```
     <assertedIdentity="&quot;Super Admin&quot; <glm:GLMSuperAdmin>"/>
     ```

     instead of

     ```
     <assertedIdentity=""Super Admin" <glm:GLMSuperAdmin>" />
     ```

   user=*""*
   : A user ID with administrator-level privileges, for connecting to the Aggregation Proxy. This value is optional.

password=""
Password of the administrator-level user, for connecting to the Aggregation Proxy. Required only when a user is specified.

subscribeExpiration="63"
Time, in minutes, to specify in the Expire header when Presence Server sends SUBSCRIBE requests to the IBM XDMS shared list server. The subscription will be refreshed as long as there are client subscriptions on the server.

retryInterval="5"
Time, in minutes, to wait before retrying a request that fails. This applies to outgoing SUBSCRIBE requests sent to an external source–for example, the IBM XDMS shared list server. Presence Server sends the requests to subscribe to the group content and to receive notification when the group content is modified. The valid range is 1-1440.

xcapRoot=""
HTTP URL of the IBM XDMS server from which Presence Server should get data using `xcap_get` requests. This attribute provides a way for Presence Server to connect directly with the IBM XDMS server rather than connecting through the Aggregation Proxy.

When `xcapRoot` is specified, Presence Server uses this value as the URI for performing the `xcap_get` operation–bypassing the Aggregation Proxy. In this case, Presence Server uses the asserted identity to perform the operation.

When `xcapRoot` is not specified, Presence Server performs the `xcap_get` operation using the `xcap root` URI that is specified in the NOTIFY request received from IBM XDMS. In this case, Presence Server uses the specified user ID and password to perform the operation.

You must specify an `xcapRoot` when `enableXcapEvent` is true.

whiteListPath=""
The IBM XDMS path that designates a white list–a list of presentities whose requests are always authorized.

blackListPath=""
The IBM XDMS path that designates a black list–a list of presentities whose requests are never authorized.

specifyAssertedIdentityOnAllRequests="true"
If set to `true`, a P-Asserted-Identity must be specified in all outgoing subsequent subscribe requests. If set to `false`, a P-Asserted-Identity is required only in the first outgoing SUBSCRIBE request but is not required in subsequent SUBSCRIBE requests.

For example:

```
<authorizationLists enable="true"
sipAddress="sip:xdmshostname.example.com:5070"
fromURI="sip:user@example.com"
assertedIdentity="sip:superadmin@example.com" user="glsuser"
                 password="authpw" subscribeExpiration="63"
retryInterval="1" xcapRoot=""
whiteListPath="xcap.example.com/services/resource-lists/users/
sip:white.xml"                  blackListPath="xcap.example.com/
```

```
services/resource-lists/users/sip:black.xml"
specifyAssertedIdentityOnAllRequests="true"/>
```

4. Modify the attributes on the `publicIdMapping` tag to define the ways in which Presence Server uses mapping and normalization for presentities on white and black lists, and for presence rules.

   You can set the following attributes. All values are disabled by default.

   enableForWhiteList
   `true` to enable ID mapping for presentities on the white list; `false` to disable.

   enableForBlackList
   `true` to enable ID mapping for presentities on the black list; `false` to disable.

   enableForPresenceRules
   `true` to enable the testing of identity conditions for comparison in Presence rules documents; `false` to disable.

   For example:

   ```
   <publicIDMapping enableForWhiteList="true" enableForBlackList="true"
   enableForPresenceRules="true"/>
   ```

5. Modify the attributes on the `presenceRules` tag to specify an IBM XDMS server and user name for getting Presence rules documents.

   You can set the following attributes. Default values are shown.

   enable=*"false"*
   `true` to enable IBM XDMS to get presence rules; `false` to disable

   XDMSSipAddress=*""*
   SIP Address, port, and transport protocol (optional) of the IBM XDMS server to which subscription requests are sent

   fromUri
   A valid SIP URI that will appear in the From: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. This value is required.

   assertedIdentity=*""*
   URI, in scheme:identity format, of the requester authorized to get XCAP documents

   This value will appear in the P-Asserted-Id: header for outgoing SUBSCRIBE requests sent to the IBM XDMS shared list server. The identity must have IBM XDMS administrator authority to access user groups.

   For the `assertedIdentity` attribute, the display name is optional. If you choose to define a display name, you must use the HTML code for symbols for the value of the attribute. For example, use

   ```
   <assertedIdentity="&quot;Super Admin&quot; &lt;glm:GLMSuperAdmin&gt;"/>
   ```

   instead of

   ```
   <assertedIdentity=""Super Admin" <glm:GLMSuperAdmin>" />
   ```

   user=*""*
   A user ID with administrator-level privileges, for connecting to the Aggregation Proxy. This value is optional.

password=""
Password of the administrator-level user, for connecting to the Aggregation Proxy. Required only when a user is specified.

subscribeExpiration="63"
Time, in minutes, to specify in the Expire header when Presence Server sends SUBSCRIBE requests to the IBM XDMS shared list server. The subscription will be refreshed as long as there are client subscriptions on the server.

retryInterval="5"
Time, in minutes, to wait before retrying a request that fails. This applies to outgoing SUBSCRIBE requests sent to an external source–for example, the IBM XDMS shared list server. Presence Server sends the requests to subscribe to the group content and to receive notification when the group content is modified. The valid range is 1-1440.

xcapRoot=""
HTTP URL of the IBM XDMS server from which Presence Server should get data using `xcap_get` requests. This attribute provides a way for Presence Server to connect directly with the IBM XDMS server rather than connecting through the Aggregation Proxy.

When `xcapRoot` is specified, Presence Server uses this value as the URI for performing the `xcap_get` operation–bypassing the Aggregation Proxy. In this case, Presence Server uses the asserted identity to perform the operation.

When `xcapRoot` is not specified, Presence Server performs the `xcap_get` operation using the `xcap root` URI that is specified in the NOTIFY request received from IBM XDMS. In this case, Presence Server uses the specified user ID and password to perform the operation.

You must specify an `xcapRoot` when `enableXcapEvent` is true.

enableMultipleIDMapping="true"
`true` to omit the schema name from the authorized user's URI when getting presence rules from the IBM XDMS server. `false` to include the schema name.

For example, if the user's URI is `sip:name@host`, specify `true` to identify the user to the server as `name@host`. Specify `false` to identify the user as `sip:name@host`.

6. Modify the `authorizationChangeJMS` tag to specify whether, following a change in authorization, the JMS message sent to servers in the cluster should contain the authorization document.

   Set `includeFullDocument="`*`true`*`"` (the default) to include the authorization document in the JMS message after a change in authorization.

   Set `includeFullDocument="`*`false`*`"` to omit the authorization document from the JMS message.

   A setting of `true` might improve performance because the documents are cached and do not need to be retrieved from the database. If you do not require the additional content, you can disable this feature. Doing so reduces the size of JMS messages and may, as a result, improve the performance of your system. For example:

   ```
   <authorizationChangeJMS includeFullDocument="true"/>
   ```

7. On the `authentication` tag, set `authenticatedUserOnAllRequests=true` to specify that all incoming Subscribe requests must be authenticated.

   The default value is `true`.

   When you are using the IMS Trust Association Interceptor (TAI), this setting means that a p-asserted-identity header must be specified in each incoming request.

   When `authenticatedUserOnAllRequests=false`, only the first request is authenticated. Subsequent requests in the Subscribe dialog are treated as if they came from the sender of the first request. For example:

   ```
   <authentication authenticatedUserOnAllRequests="true"/>
   ```

8. Save and close the file.
9. Open `ConfigurationParams.txt` with a text editor.
10. Update the following parameters for your environment:

    cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

    username = *database_administrator_user_name*

    password = *database_administrator_password*

    DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

    Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

    DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

    Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*

11. Run the `java` command that is appropriate for your operating system:

    Important: Enter the following parameters on a single line.

    AIX Linux
    ```
    java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig
    config_path/ConfigurationParams.txt
    ```

    (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

    Important: JDBC drivers must be separated by a colon.
    For example:

    AIX
    ```
    /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
    UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
    db2/V9.5/java/db2jcc_license_cu.jar CmdConfig
    ConfigurationParams.txt
    ```

    Linux
    ```
    /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
    UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
    db2/V9.5/java/db2jcc_license_cu.jar CmdConfig
    ConfigurationParams.txt
    ```

12. Restart the application:
    a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

       Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.

    b. Select the check box associated with the Presence Server.

c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
d. Click **Start**. The **Application Status** column should indicate a `Started` status.

13. Restart all servers that are running applications that have implemented the authorization service.

Results

Note: `xcap_get` requests for external lists group lists uses user and password, if available. If there is no configured user, asserted identity is used `assertedIdentity`.

Configuring the way in which PUBLISH and SUBSCRIBE requests are handled

Presence Server has configuration options that determine the way in which PUBLISH and SUBSCRIBE requests are handled.

Configuring partial publish and partial notify

You can use the configuration file to enable and disable partial publish and partial notify.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the attributes on the `partialDocuments` tag to enable or disable partial publish and partial notify.

   You can set the following attributes. By default, both partial publish and partial notify are disabled.

   acceptPartialPublish=""
   `true` to enable partial publish; `false` to disable it

   sendPartialNotify=""
   `true` to enable partial notify; `false` to disable it

   For example:

   ```
   <partialDocuments acceptPartialPublish="true"
   sendPartialNotify="true"/>
   ```

3. Save and close the file.
4. Open `ConfigurationParams.txt` with a text editor.
5. Update the following parameters for your environment:

cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

username = *database_administrator_user_name*

password = *database_administrator_password*

DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*

6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX
   ```
   /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

   Linux
   ```
   /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.

   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring expiration times for PUBLISH and SUBSCRIBE requests

You can assign default values for the expiration times of PUBLISH and SUBSCRIBE requests by modifying their values in the configuration file.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`

- ConfigurationParams.txt
- UpdateConfiguration.jar

Before you can make changes to your configuration, the file UpdateConfiguration.jar must be in the class path.

About this task

The expiration time for any PUBLISH or SUBSCRIBE request is determined by the request's Expires header. You can configure Presence Server to assign a default expiration time when a request does not have an Expires header. You can also define a maximum expiration time, used when the value in the Expires header exceeds a certain threshold, and a minimum time, below which requests are rejected.

Expiration times are measured in minutes. To change the settings of expiration times for PUBLISH and SUBSCRIBE requests, complete the following steps:

1. Open SystemConfiguration.xml with a text editor.
2. Modify the attributes for the publishExpiration and subscribeExpiration tags. These tags, respectively, set the expiration times for PUBLISH requests and SUBSCRIBE requests.

   You can set the following attributes for both tags. Default values are shown.

   default=*"60"*
   The expiration time in minutes, when a PUBLISH or SUBSCRIBE request arrives with no Expires header.

   maximum=*"1440"*
   An upper limit on the expiration time in minutes, for PUBLISH and SUBSCRIBE requests. When a request arrives with a higher value specified in its Expires header, its expiration time is reset to this value. 1440 minutes, or 24 hours, is the highest value allowed.

   minimum=*"1"*
   A lower limit on the expiration time, in minutes, for PUBLISH and SUBSCRIBE requests. When a request arrives with a lower value specified in its Expires header, the request is rejected. 1 minute is the lowest value allowed.

   For example:

   ```
   <publishExpiration default="60" maximum="1440" minimum="1"/>
   <subscribeExpiration default="60" maximum="1440" minimum="1"/>
   ```

3. Save and close the file.
4. Open ConfigurationParams.txt with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for SystemConfiguration.xml)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

Oracle **dbConnectionString** =
jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*

6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX `/usr/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

   Linux `/opt/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.

   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Adding headers to outgoing NOTIFY requests

You can configure Presence Server to copy additional headers from incoming SUBSCRIBE requests to NOTIFY requests on the same session.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

About this task

The 3GPP specification for SIP requires that an application server be able to extract certain types of headers from incoming SUBSCRIBE requests and insert them into the corresponding NOTIFY requests. For example, when an application server acting as an originating user agent (UA) issues a SUBSCRIBE request and receives

a NOTIFY request, it expects to collect and use information from the P-Charging-Vector header that is included in the request.

You can configure Presence Server so that it includes copies specific types of headers from SUBSCRIBE requests and includes them in the NOTIFY requests it sends.

To arrange for one or more header types to be copied and included in NOTIFY requests, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. In the `<copyToNotifyHeaders>` element, set `enable="true"`.
3. Within the `<copyToNotifyHeaders>` element, create one or more `header` tags–one for each header type that you want to configure.

   You can set the following attribute for `header`. There is no default value.

   name=*"header_type"*
   Name of the header type, as specified in 3GPP specification 24.229 for SIP.

   For example:

   ```
   <copyToNotifyHeaders enable="true">
   <header name="P-Charging-Vector" />
   </copyToNotifyHeaders>
   ```

4. Save and close the file.
5. Open `ConfigurationParams.txt` with a text editor.
6. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

   Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*
7. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX
   ```
   /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

Linux /opt/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt

8. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring content indirection

You can configure content indirection in SIP messages, allowing the messages to contain indirect references to specific content. Content indirection is enabled by default.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

Be sure that the database is running before you make changes to the configuration.

About this task

Follow these instructions to configure content indirection.

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the attributes on the `contentIndirection` tag to enable or disable content indirection.

   You can set the following attributes. By default, content indirection is disabled.

   enable=""
   `true` to enable content indirection; `false` to disable it.

   maxDocumentLength=""
   Maximum size, in bytes, of the message body. When a message is larger than this size, content indirection is used. The valid range is 500-5000. The default value is 1000.

   proxyHttpUrl
   HTTP URL for the content indirection Servlet. The host name should be the name of a SIP application server in the stand-alone environment or the name of a proxy host in the clustered environment. This URL will be returned in NOTIFY messages.

documentExpiration
Time, in seconds, that the content indirection document should remain available. The valid range is 60-36000. The default value is 600.

For example:

```
<contentIndirection enable="true" maxDocumentLength="1000"
                    proxyHttpUrl="http://sip.server.example.com:9080/
siplets/content" documentExpiration="600" />
```

3. Save and close the file.
4. Open `ConfigurationParams.txt` with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

   Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*
6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX
   ```
   /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

   Linux
   ```
   /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```
7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

8. Restart all servers that are running applications that have implemented content indirection service.

Configuring routing using the Presence Server entry point

Use the configuration file to assign different Presence Server clusters for handling requests from specified groups of users, and to specify the mode in which the Presence Server application should operate (RLS stand-alone mode, Presence Server stand-alone mode, or both modes simultaneously.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*/installableApps/presence/scripts/config.

- SystemConfiguration.xml
- ConfigurationParams.txt
- UpdateConfiguration.jar

Before you can make changes to your configuration, the file UpdateConfiguration.jar must be in the class path.

About this task

To route requests to specific Presence Server clusters, and to configure for Presence Server stand-alone mode or RLS stand-alone mode, complete the following steps.

When you enable the routing service, you should also configure Presence Server to run in RLS mode. Do not enable Presence Server stand-alone mode.

Refer to the topic *The Presence Server entry point* for additional information about these product features.

1. Open SystemConfiguration.xml with a text editor.
2. Modify the attributes on the routingService tag to configure routing.
   a. Set enable="true" to enable the routing service. By default, the routing service is disabled.
   b. Use the class attribute to specify the Java routing class implementation. This class should implement com.ibm.presence.api.routingRoutingService.

      The Presence Server product includes a default implementation for the routing service: com.ibm.presence.routing.DefaultRoutingService. When you use this default routing implementation, you will need to configure the Presence Server standalone address.
   c. Code presenceServerCluster tags inside the routingService tag. Include a presenceServerCluster tag for each stand-alone instance of Presence Server.

      The presenceServerCluster tag must have an address attribute containing a valid SIP URI for the specific Presence Server cluster to which requests are to be forwarded.

   For example:

```
<routingService enable="true"
class="com.ibm.presence.routing.DefaultRoutingService">
<presenceServerCluster address="sip:pshostname.example.com:7010"/>
</routingService>
```

3. Modify the attributes on the `standaloneRLS` tag to specify that the Presence Server application should run in RLS stand-alone mode.

   You can set the following attributes.

   enable=""
   `true` to run in RLS mode; `false` otherwise. The default value is `false`.

   Note: When you enable RLS mode, the routing component (`routingService`) must also be enabled as described in step 2 on page 95.

   entryAddress
   A valid SIP URI representing the Presence Server deployment entry point. For example, this address can be the SIP address for the S-CSCF server.

   assertedIdentity
   A valid SIP URI representing the RLS SIP user ID to use when sending back-end subscriptions to clusters that are running in Presence Server stand-alone mode.

   For example:

   ```
   <standaloneRLS enable="true"
   entryAddress="sip:scscfhostname.example.com:7010;transport=TCP"
                    assertedIdentity="sip:superadmin@example.com"/>
   ```

4. Modify the attributes on the `standalonePresenceServer` tag to specify that the Presence Server application should run in Presence Server stand-alone mode.

   You can set the following attributes.

   enable=""
   `true` to run in Presence Server stand-alone mode; `false` otherwise. If you enabled the routing service in step 2 on page 95, you must specify `false`. The default value is `false`.

   RLSAssertedIdentity
   A valid SIP URI to be used as the asserted identity for the stand-alone Presence Server and as a "From" header in the back-end subscription that is created by the RLS and forwarded to Presence Server. Should be the same as `assertedIdentity` in the `standaloneRLS` tag.

   For example:

   ```
   <standalonePresenceServer enable="true"
   RLSAssertedIdentity="sip:superadmin@example.com"/>
   ```

5. Save and close the file.
6. Open `ConfigurationParams.txt` with a text editor.
7. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   dbDriver = com.ibm.db2.jcc.DB2Driver

   dbDriver = oracle.jdbc.driver.OracleDriver

   dbConnectionString = jdbc:db2://*database_host_name:database_port/database_name*

Oracle **dbConnectionString** =
jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*

8. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX `/usr/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

   Linux `/opt/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

9. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring watcher information

Presence Server has configuration options that determine the way in which watcher information is handled.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

Be sure that the database is running before you make changes to the configuration.

About this task

Watcher information consists of the current state for all subscriptions to a particular resource. Monitoring watcher information helps you control access to a resource because it provides the raw data you need to create and deploy presence authorization rules.

For more information about watcher information, refer to IETF RFC 3857 and RFC 3858.

To configure the monitoring of watcher information, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the attributes on the `watcherInfo` tag to configure the way in which watcher information is monitored.

   You can set the following attributes. Default values are shown.

   enable=*"false"*
   `true` to enable monitoring of watcher information; `false` to disable it.

   waitingDays=*"4"*
   Number of days to monitor subscriptions that are in the waiting state.

   waitingCheckInterval=*"60"*
   Time, in minutes, between cleanup operations for waiting subscriptions that should no longer be monitored. A waiting subscription should not be monitored when it is older than the value specified in `waitingDays`. The cleanup operation checks for waiting subscriptions that should not be monitored and removes any that are found.

   For example:

   ```
   <watcherInfo enable="true" waitingDays="4"
   waitingCheckInterval="60"/>
   ```

3. Save and close the file.
4. Open `ConfigurationParams.txt` with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

   Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*
6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

(where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

Important: JDBC drivers must be separated by a colon.
For example:

AIX `/usr/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

Linux `/opt/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring performance-related settings

You can use the Presence Server configuration file to activate key performance indicators and specify other settings to optimize your system's performance.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

About this task

Presence Server offers the following performance-related configuration settings:

- You can activate key performance indicators (KPI) to trace system activity. These performance indicators are based on the Performance Monitor Infrastructure (PMI) found in WebSphere Application Server, and you can view them.
- When a presence document changes (as the result of a PUBLISH request or expiration), the server publishes a JMS message to indicate the change. Additional content–for example, the modified document and the document version number--is included so that other servers can retrieve the documents more easily. If you do not require the additional content, you can disable this feature. Doing so reduces the size of JMS messages and may, as a result, improve the performance of your system.

- When Presence Server receives a presence rules authorization document, it sends a JMS message to other servers in the cluster. The authorization document is included in the JMS message by default. You can override this default to improve performance when working presence rules authorization is enabled.
- The notification throttling mechanism allows client applications to limit the rate of notifications, to decrease the processing overhead and the network load. A significant saving may be achieved with subscriptions to resources with high notification rate, such as subscriptions to large resource lists.
- You can configure Presence Server so that it waits to send the first NOTIFY message until it has received all of the information from a subscriber (IBM XDMS or an external source). When notification throttling is enabled, this setting can prevent the sending of incomplete NOTIFY messages. It can also improve performance because it reduces the number of NOTIFY messages sent to the subscriber.
- You can also disable the caching of changed documents. Doing this can reduce the number of database transactions, resulting in improved performance.

To activate and deactivate performance-related tools, follow these steps. The topic *Performance metrics* lists the key performance indicators that are supported by Presence Server .

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify the attributes on the `PMI` tag to activate or deactivate PMI-based key performance indicators (KPIs).

   You can set the following attributes. Default values are shown.

   accumulationTime=*"1"*
   : The amount of time, in minutes, for which KPI data is collected. Data older than the specified time interval is discarded.

   outputToTrace=*"false"*
   : `true` to store KPI data in trace records so that it can be viewed in logs; `false` not to store it in trace records.

   outputViaSOA=*"false"*
   : `true` to store KPI data in a format where it can be viewed using service-oriented architecture (SOA) software; `false` not to store it in this fashion.

   For a complete list of the PMI-based key performance indicators that are used by Presence Server, refer to the topic *Key performance indicators*.
3. Modify the `documentChangeJMS` tag to specify whether the contents of changed documents should be included in JMS messages.

   Set `includeFullDocument="`*`true`*`"` (the default) to include the contents of changed documents in JMS messages.

   Set `includeFullDocument="`*`false`*`"` to omit the contents of changed documents from the messages.

   A setting of `true` might improve performance because the documents are cached and do not need to be retrieved from the database. For example:

   ```
   <documentChangeJMS includeFullDocument="true"/>
   ```
4. Modify the `authorizationChangeJMS` tag to specify whether, following a change in authorization, the JMS message sent to servers in the cluster should contain the authorization document.

   Set `includeFullDocument="`*`true`*`"` (the default) to include the authorization document in the JMS message after a change in authorization.

Set includeFullDocument="*false*" to omit the authorization document from the JMS message.

A setting of true might improve performance because the documents are cached and do not need to be retrieved from the database. If you do not require the additional content, you can disable this feature. Doing so reduces the size of JMS messages and may, as a result, improve the performance of your system. For example:

```
<authorizationChangeJMS includeFullDocument="true"/>
```

5. Modify the attributes on the throttling tag to limit the rate of SIP event notifications for one or more event packages.

   Set enable="true" to enable notification throttling; set enable="false" to disable it. By default, notification throttling is enabled.

   You can set throttling intervals for three types of event packages. Within the throttling tag, each event package is represented by a separate event tag with the following attributes:

   name=""
   String specifying the type of event package. The following values are valid:
   - presence
   - presence.winfo
   - presence.winfo.winfo

   default=""
   Default throttle interval, in seconds. A SIP notification for the specified event package will be accepted only once during this interval. The valid range is 0-3600. The default value is 5.

   minimum=""
   Minimum throttle interval, in seconds. The valid range is 0-3600. The default value is 5.

   maximum=""
   Maximum throttle interval, in seconds. The valid range is 0-3600. The default value is 600.

   For example:

```
<throttling enable="true">      <event name="presence" default="5"
minimum="5" maximum="600"/>      <event name="presence.winfo"
default="5" minimum="5" maximum="600"/>      <event
name="presence.winfo.winfo" default="5" minimum="5" maximum="600"/>
</throttling>
```

6. On the waitForExternalInformation tag, modify the maxInterval attribute to specify how long to wait while collecting information from IBM XDMS and other external sources before notifying the client.

   maxInterval specifies the time, in seconds, to wait while collecting information from IBM XDMS and other external sources before sending a notification to the client. For group subscriptions, the time is measured after the group content is received from IBM XDMS.

   Valid values are 0-60. The default value is 15.

   Note: Avoid setting the interval too small, to avoid having notifications sent before all information has been collected. It is also recommended that

you select a value of 30 or below because a subscriber may invalidate the subscription if a notification is not returned within 30 seconds.

For example:

```
<waitForExternalInformation maxInterval="15" />
```

7. Modify the `documentCache` tag to specify whether the contents of changed documents are saved in cache.

   Set `enable="true"` (the default) to cache the contents of changed documents.

   Set `enable="false"` to avoid caching changed documents.

   A setting of `true` might improve performance because the documents are cached and therefore do not need to be retrieved from the database. For example:

   ```
   <documentCache enable="true"/>
   ```

8. Save and close the file.
9. Open `ConfigurationParams.txt` with a text editor.
10. Update the following parameters for your environment:

    cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

    username = *database_administrator_user_name*

    password = *database_administrator_password*

    DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

    Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

    DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

    Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*

11. Run the `java` command that is appropriate for your operating system:

    Important: Enter the following parameters on a single line.

    AIX Linux
    ```
    java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt
    ```

    (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

    Important: JDBC drivers must be separated by a colon.
    For example:

    AIX
    ```
    /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
    UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
    db2/V9.5/java/db2jcc_license_cu.jar CmdConfig
    ConfigurationParams.txt
    ```

    Linux
    ```
    /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
    UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
    db2/V9.5/java/db2jcc_license_cu.jar CmdConfig
    ConfigurationParams.txt
    ```

12. Restart the application:
    a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.

b. Select the check box associated with the Presence Server.

c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.

d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring usage records

Presence Server can be configured to log usage records for specific SIP requests.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

Be sure that the database is running before you make changes to the configuration.

About this task

Usage records allow you to capture usage information about PUBLISH, SUBSCRIBE, and NOTIFY requests.

To configure usage records, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. Modify attributes on the `usageRecords` tag to configure whether usage records are captured for each type of SIP request (PUBLISH, SUBSCRIBE, and NOTIFY).

   The default settings are as follows:

   ```
   publish="true"
   subscribe="true"
   notify="true"
   ```

   For example:

   ```
   <usageRecords publish="true" subscribe="true" notify="true"/>
   ```
3. Save and close the file.
4. Open `ConfigurationParams.txt` with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

DB2 **dbConnectionString** = jdbc:db2://*database_host_name:database_port/database_name*

Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name:database_port:database_name*

6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX `/usr/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

   Linux `/opt/IBM/WebSphere/AppServer/java/bin/java -classpath UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt`

7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.

   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring error notification

You can configure Presence Server to send notifications to Java Management Extensions (JMX) services when error conditions occur.

Before you begin

You will need the following files to complete this task. After you install the Presence Server product, these files are found in the directory *was_root*`/installableApps/presence/scripts/config`.

- `SystemConfiguration.xml`
- `ConfigurationParams.txt`
- `UpdateConfiguration.jar`

Before you can make changes to your configuration, the file `UpdateConfiguration.jar` must be in the class path.

It is assumed that you have configured JMX services, using WebSphere Application Server, to receive the error notifications sent by Presence Server.

About this task

To enable notification for JMX services, complete the following steps:

1. Open `SystemConfiguration.xml` with a text editor.
2. Supply values for the attributes on the `JmxListener` tag to designate a WebSphere Application Server user ID to be used to notify JMX services when error conditions occur while Presence Server is running.

   You can set the following attributes. Default values are shown.

   enable="false"
   `true` to enable the sending of error notifications to JMX services; `false` to disable.

   user=""
   The WebSphere Application Server user on whose behalf notifications are to be sent

   password=""
   The user's password.
3. Save and close the file.
4. Open `ConfigurationParams.txt` with a text editor.
5. Update the following parameters for your environment:

   cfg.system = *xml_path* (where *xml_path* is the directory location for `SystemConfiguration.xml`)

   username = *database_administrator_user_name*

   password = *database_administrator_password*

   DB2 **dbDriver** = com.ibm.db2.jcc.DB2Driver

   Oracle **dbDriver** = oracle.jdbc.driver.OracleDriver

   DB2 **dbConnectionString** = jdbc:db2://*database_host_name*:*database_port*/*database_name*

   Oracle **dbConnectionString** = jdbc:oracle:thin:@*database_host_name*:*database_port*:*database_name*
6. Run the `java` command that is appropriate for your operating system:

   Important: Enter the following parameters on a single line.

   AIX Linux `java -classpath UpdateConfiguration.jar: jdbc_path CmdConfig config_path/ConfigurationParams.txt`

   (where *jdbc_path* is the directory location for your JDBC drivers and *config_path* is the directory location for `ConfigurationParams.txt`)

   Important: JDBC drivers must be separated by a colon.
   For example:

   AIX
   ```
   /usr/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

   Linux
   ```
   /opt/IBM/WebSphere/AppServer/java/bin/java -classpath
   UpdateConfiguration.jar:/opt/IBM/db2/V9.5/java/db2jcc.jar:/opt/IBM/
   db2/V9.5/java/db2jcc_license_cu.jar CmdConfig ConfigurationParams.txt
   ```

7. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Configuring the REST interface to Presence Server

Use the WebSphere Integrated Solutions Console to configure the Presence HTTP gateway, on which the REST interface to Presence Server is defined.

About this task

Note that these configuration attributes, unlike other configuration options for your Presence Server deployment, are not stored in the XML configuration file (`SystemConfiguration.xml`).

The configuration attributes for the Presence Servlet are added as context parameters of the Servlet's `web.xml` file. To set the attributes, follow these steps.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
3. In the list of applications, click the name of the Presence Server application.
4. In the Enterprise Applications page for Presence Server, under Environment entries for Web modules, assign values to the following parameters:

   PresenceServerAddress
   The SIP address of the Presence Server. The HTTP gateway will use this address to send the SIP requests after translating them from HTTP requests. In most cases, the value for this parameter will be the address of the SIP proxy that you are using. The default value is sip:localhost:5060;transport=TCP.

NotifyTimeout
The maximum time, in seconds, to wait for a NOTIFY message from the Presence Server before sending an HTTP error response (408 Request timeout) to the client. The default value is 40 seconds.

CacheHeader
The Cache-Control header value to include in responses to HTTP requests. The header is used to disable caching in the HTTP client and in proxy servers along the way to the client. The default value is as follows: private, no-cache, max-age=0, must-revalidate.

5. Optionally, change the REST application prefix–the context root–from its default value of `presence`:
   a. In the Enterprise Applications page for Presence Server, under Context root for Web modules, change the context root for the `presence.rest.web` module.
   b. Open a command prompt and navigate to the *was_root*`/plugins` directory.
   c. Open the `config.xml` in `IBMWebSpherePresenceServerRESTPlugin.jar` and update the `contextRoot` element.
   d. Save your changes.
6. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

The XML configuration file

An XML file is used to set up the configuration during installation. The same file is used to update the configuration as needed.

Configuration file location

During the installation process, the XML configuration file–`SystemConfiguration.xml`–is copied to thePresence Server. The default location is

AIX Linux *was_root*`/installableApps/presence/scripts/config/SystemConfiguration.xml`

Note: *was_root* is the installation root directory for WebSphere Application Server Network Deployment. By default, this directory is:

AIX `/usr/IBM/WebSphere/AppServer`

Linux `/opt/IBM/WebSphere/AppServer`

Default configuration file

The following is an example of the Presence Server SystemConfiguration.xml configuration file. The values in this file are the default values that Presence Server uses if a configuration file is not found in the database.

```
<system-configuration version="7.0">
 <!-- Expiration period (in minutes) of SIP PUBLISH requests -->
 <publishExpiration default="60" maximum="1440" minimum="1"/>

 <!-- Expiration period (in minutes) of SIP SUBSCRIBE requests -->
 <subscribeExpiration default="60" maximum="1440" minimum="1"/>

 <!--  Usage records generation per SIP request -->
 <usageRecords publish="true" subscribe="true" notify="true"/>

 <!--  PMI configuration -->
 <PMI accumulationTime="1" outputToTrace="false" outputViaSOA="false"/>

 <!-- Flag indicating whether the JMS message sent to servers in the cluster following a change in a
 should include the full document in the JMS message body -->
 <documentChangeJMS includeFullDocument="true"/>

 <!-- Cache Configuration -->
 <documentCache enable="true"/>

 <!-- Partial Configuration -->
 <partialDocuments acceptPartialPublish="false" sendPartialNotify="false"/>

 <!-- Configuring XDMS server, the sip-address for SUBSCRIBE request, and authorized user to get XCAI
 <groupListServer enable="false" sipAddress="" fromURI="" assertedIdentity="" user="" password="" sul
                  xcapRoot="" enableXcapEvent="" specifyAssertedIdentityOnAllRequests="true"/>

 <!-- Configuring S-CSCF server, the sip-address to send SUBSCRIBE requests is required -->
 <S-CSCF enable="false" sipAddress="" fromURI="" assertedIdentity="" subscribeExpiration="63"
   retryInterval="5" specifyAssertedIdentityOnAllRequests="true"/>

 <!-- Watcher information subscriptions -->
 <watcherInfo enable="false" waitingDays="4" waitingCheckInterval="60"/>

 <!--  disabling/enabling the authorization mechanism  -->
 <externalAuthorization enable="false"/>

 <!-- white and black list authorization  -->
 <authorizationLists enable="false" sipAddress="" fromURI="" assertedIdentity="" user="" password=""
                  xcapRoot="" whiteListPath="" blackListPath="" specifyAssertedIdentityOnAllRequests=

 <!-- Configuring XDMS server, the sip-address for SUBSCRIBE request, and authorized user to get pres
 <presenceRules enable="false" XDMSSipAddress="" fromURI="" assertedIdentity="" user="" password="" s
                  xcapRoot="" enableMultipleIDMapping="true" specifyAssertedIdentityOnAllRequests="t

 <!-- Flag indicating whether the JMS message sent to servers in the cluster following a change in au
   should include the authorization document in the JMS message body  -->
 <authorizationChangeJMS includeFullDocument="true"/>

 <!-- list of SIP external sources for presence informations  -->
 <externalSources>
  <externalSource enable="false" sipAddress="" fromURI="" assertedIdentity="" subscribeExpiration="60
                  triggerEventPackage="" sourceEventPackage="" acceptHeader="" specifyAssertedIdenti
 </externalSources>

 <!-- JmxListener user and password, for Mbean server connection -->
 <jmxListener enable="false" user="" password=""/>

 <!--  Configure the presence server to copy headers from incoming Subscribe requests to Notify requ
 <copyToNotifyHeaders enable="false">
    <header name="" />
```

```
</copyToNotifyHeaders>

<!--  Configure Presence Server to consider fundamental URIs for presentity management.  -->
   <publicIDMapping enableForWhiteList="false" enableForBlackList="false" enableForPresenceRules=

<!--  Configure how long to wait (in seconds) while collecting information from XDMS and external
     sending the notification to the client  -->
<waitForExternalInformation maxInterval="15" />

<!--  Configure Notification throttling -->
<throttling enable="true">
 <!-- list of event package and default throttle intervals (in seconds) -->
 <event name="presence" default="5" minimum="5" maximum="600"/>
 <event name="presence.winfo" default="5" minimum="5" maximum="600"/>
 <event name="presence.winfo.winfo" default="5" minimum="5" maximum="600"/>
</throttling>

<!-- Configure Content Indirection -->
<contentIndirection enable="false" maxDocumentLength="1000" proxyHttpUrl="" documentExpiration="6

<!-- Standalone Presence Server   -->
<standalonePresenceServer enable="false" RLSAssertedIdentity=""/>

<!-- Standalone RLS configuration  -->
<standaloneRLS enable="false" entryAddress="" assertedIdentity=""/>

<!-- Routing service configuration  -->
<routingService enable="false" class="com.ibm.presence.routing.DefaultRoutingService">
 <presenceServerCluster address=""/>
 <presenceServerCluster address=""/>
 <presenceServerCluster address=""/>
</routingService>

<!-- Configure Presence Server authentication behavior -->
<authentication authenticatedUserOnAllRequests="true"/>

</system-configuration>
```

For the assertedIdentity attribute, the display name is optional. If you choose to define a display name, you must use the HTML code for symbols for the value of the attribute. For example, use

```
<assertedIdentity="&quot;Super Admin&quot; &lt;glm:GLMSuperAdmin&gt;"/>
```

instead of

```
<assertedIdentity=""Super Admin" <glm:GLMSuperAdmin>" />
```

Sample configuration file

The following sample SystemConfiguration.xml file includes sample values to help you understand the syntax of the attributes.

```
<system-configuration version="7.0">
 <!-- Expiration period (in minutes) of SIP PUBLISH requests -->
 <publishExpiration default="60" maximum="1440" minimum="1"/>

 <!-- Expiration period (in minutes) of SIP SUBSCRIBE requests -->
 <subscribeExpiration default="60" maximum="1440" minimum="1"/>

 <!--  Usage records generation per SIP request -->
 <usageRecords publish="true" subscribe="true" notify="true"/>

 <!--  PMI configuration -->
 <PMI accumulationTime="1" outputToTrace="false" outputViaSOA="false"/>

 <!-- Flag indicating whether the JMS message sent to servers in the cluster following a change ir
```

```
should include the full document in the JMS message body -->
<documentChangeJMS includeFullDocument="true"/>

<!-- Cache Configuration -->
<documentCache enable="true"/>

<!-- Partial Configuration -->
<partialDocuments acceptPartialPublish="true" sendPartialNotify="true"/>

<!-- Configuring XDMS server, the sip-address for SUBSCRIBE request, and authorized user to get XCAI
<groupListServer enable="true" sipAddress="sip:server.example.com:7010;transport=UDP"
                 assertedIdentity="&quot;Super Admin1&quot;&lt;sip:ps@example.com&gt;" user="glsuser
                 retryInterval="5" subscribeExpiration="63" xcapRoot="http://xdmshostname.example.co
                 enableXcapEvent="true" specifyAssertedIdentityOnAllRequests="true"/>

<!-- Configuring S-CSCF server, the sip-address to send SUBSCRIBE requests is required -->
<S-CSCF enable="true" sipAddress="sip:server.example.com:7010;transport=UDP"
                 fromURI="sip:Admin1@example.com" assertedIdentity="&quot;Super Admin1&quot;&lt;sip
                 subscribeExpiration="63" retryInterval="5" specifyAssertedIdentityOnAllRequests="tr

<!-- Watcher information subscriptions -->
<watcherInfo enable="true" waitingDays="4" waitingCheckInterval="60"/>

<!--  disabling/enabling the authorization mechanism  -->
<externalAuthorization enable="true"/>

<!-- white and black list authorization  -->
<authorizationLists enable="true" sipAddress="sip:xdmshostname.example.com:5070"
                 fromURI="sip:user@example.com" assertedIdentity="sip:superadmin@example.com" user="
                 password="authpw" subscribeExpiration="63" retryInterval="1" xcapRoot=""
                 whiteListPath="xcap.example.com/services/resource-lists/users/sip:white.xml"
                 blackListPath="xcap.example.com/services/resource-lists/users/sip:black.xml"  spec

<!-- Configuring XDMS server, the sip-address for SUBSCRIBE request, and authorized user to get pres
<presenceRules enable="true" XDMSSipAddress="sip:xdmshostname.example.com:5070" assertedIdentity="s
                 fromURI="sip:user@example.com" xcapRoot="http://xdmshostname.example.com:9086/serv
                 user="glsuser" password="authpw" subscribeExpiration="63" retryInterval="5"
                 enableMultipleIDMapping="true" specifyAssertedIdentityOnAllRequests="true"/>
<!-- Flag indicating whether the JMS message sent to servers in the cluster following a change in au
  should include the authorization document in the JMS message body  -->
<authorizationChangeJMS includeFullDocument="true"/>
<!-- list of SIP external sources for presence informations  -->
<externalSources>
 <externalSource enable="true" sipAddress="sip:server.example.com:7010;transport=UDP"
                 fromURI="sip:Admin1@example.com" assertedIdentity="&quot;Super Admin1&quot;&lt;sip
                 subscribeExpiration="20" retryInterval="10" triggerEventPackage="presence" sourceEv
                 acceptHeader="" specifyAssertedIdentityOnAllRequests="true"/>
</externalSources>

<!-- JmxListener user and password, for Mbean server connection -->
<jmxListener enable="true" user="wasuser" password="passwd"/>

<!--  Configure the presence server to copy headers from incoming Subscribe requests to Notify requ
<copyToNotifyHeaders enable="true">
   <header name="P-Charging-Vector" />
</copyToNotifyHeaders>

<!--  Configure Presence Server to consider fundamental URIs for presentity management.  -->
<publicIDMapping enableForWhiteList="true" enableForBlackList="true" enableForPresenceRules="true"/>

<!--  Configure how long to wait (in seconds) while collecting information from XDMS and external so
  sending the notification to the client  -->
<waitForExternalInformation maxInterval="15" />

<!--  Configure Notification throttling -->
<throttling enable="true">
     <event name="presence" default="5" minimum="5" maximum="600"/>
```

```
     <event name="presence.winfo" default="5" minimum="5" maximum="600"/>
     <event name="presence.winfo.winfo" default="5" minimum="5" maximum="600"/>
 </throttling>

 <!-- Configure Content Indirection -->
 <contentIndirection enable="true" maxDocumentLength="1000"
                 proxyHttpUrl="http://sip.server.example.com:9080/siplets/content" documentExpira

 <!-- Standalone Presence Server   -->
 <standalonePresenceServer enable="true" RLSAssertedIdentity="sip:superadmin@example.com"/>

 <!-- Standalone RLS configuration  -->
 <standaloneRLS enable="true" entryAddress="sip:scscfhostname.example.com:7010;transport=TCP"
                 assertedIdentity="sip:superadmin@example.com"/>

 <!-- Routing service configuration  -->
 <routingService enable="true" class="com.ibm.presence.routing.DefaultRoutingService">

  <presenceServerCluster address="sip:pshostname.example.com:7010"/>
  </routingService>

 <!-- Configure Presence Server authentication behavior -->
 <authentication authenticatedUserOnAllRequests="true"/>

</system-configuration>
```

Chapter 5. Administering the IBM WebSphere Presence Server Component

This documentation provides instructions and information needed to administer the IBM WebSphere Presence Server Component.

Restarting Presence Server

You can restart Presence Server using the Integrated Solutions Console.

About this task

Complete the following steps to restart Presence Server.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Restart the application:
   a. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   b. Select the check box associated with the Presence Server.
   c. Click **Stop**. The **Application Status** column should indicate a `Stopped` status.
   d. Click **Start**. The **Application Status** column should indicate a `Started` status.

Stopping and starting the server

After making changes to the server configuration, you must restart the application server.

About this task

In a clustered environment, some tasks require you to restart the deployment manager for changes to take effect. To stop the deployment manager, you must stop all application servers, all node agents, and then the deployment manager. To restart the deployment manager, you must start the deployment manager, all node agents, and then the cluster (which starts all application servers).

The following instructions describe how to stop and restart resources both from the Integrated Solutions Console and from a command-line prompt.

Note: *was_profile_root* is the directory for a WebSphere Application Server Network Deployment profile called *profile_name*. By default, this directory is:

- AIX `/usr/IBM/WebSphere/AppServer/profiles/profile_name`
- Linux `/opt/IBM/WebSphere/AppServer/profiles/profile_name`

Stopping a cluster

About this task

When you stop a cluster, all application servers on the cluster are stopped.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Stop the cluster:
   a. In the Integrated Solutions Console, click **Servers** → **Clusters** → **WebSphere application server clusters**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Clusters**.
   b. Select the check box associated with the name of the cluster.
   c. Click **Stop**.

Stopping a server (console)

About this task

Stopping an application server stops all applications automatically.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.

c. Click **Log in**.

2. Stop the application server:
   a. In the Integrated Solutions Console, click **Servers** → **Server Types** → **WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
   b. Select the check box associated with the name of the server.
   c. Click **Stop**.

Stopping a server (command line)

Run the following command:

AIX *was_profile_root*/bin/stopServer.sh *server_name* -username *user_name* -password *password*

Linux *was_profile_root*/bin/stopServer.sh *server_name* -username *user_name* -password *password*

Where:

The *was_profile_root* path contains the name of the application server profile (for example, AppSrv01).

server_name is name of the application server.

user_name represents your WebSphere Application Server administrator user ID.

password represents the password associated with your *user_name*.

Stopping the node agent (console)

About this task

When stopping the deployment manager and application servers, you must also stop the node agents. If you are stopping a cluster, you must stop all node agents.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Stop one or more nodes:
   a. In the Integrated Solutions Console, click **System administration** → **Node agents**.
   b. Select the check boxes associated with each node.
   c. Click **Stop**.

Stopping the node agent (command line)

Run the following command:

AIX *was_profile_root*/bin/stopNode.sh -username *user_name* -password *password*

Linux *was_profile_root*/bin/stopNode.sh -username *user_name* -password *password*

Where:

The *was_profile_root* path contains the name of a federated node profile (for example, Custom01).

user_name represents your WebSphere Application Server administrator user ID.

password represents the password associated with your *user_name*.

Stopping the deployment manager (console)

About this task

When stopping the servers and node agents in a cluster, you must also stop the deployment manager. When the deployment manager is stopped, you will not be able to access the Integrated Solutions Console.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.
      Where:
      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.
      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Stop the deployment manager:
   a. In the Integrated Solutions Console, click **System administration** → **Deployment manager**.
   b. Click **Stop**.

Stopping the deployment manager (command line)

Run the following command:

AIX *was_profile_root*/bin/stopManager.sh -username *user_name* -password *password*

Linux *was_profile_root*/bin/stopManager.sh -username *user_name* -password *password*

Where:

The *was_profile_root* path contains the name of the deployment manager profile (for example, Dmgr01).

user_name represents your WebSphere Application Server administrator user ID.

password represents the password associated with your *user_name*.

Starting the deployment manager

About this task

Start the deployment manager before starting the node agents and application servers. When the deployment manager is started, you will have access to the Integrated Solutions Console.

Run the following command:

- AIX `was_profile_root/bin/startManager.sh`
- Linux `was_profile_root/bin/startManager.sh`

Where:

The *was_profile_root* path contains the name of the deployment manager profile (for example, Dmgr01).

Starting the node agents

Before you begin

After starting the deployment manager, you must start the node agents before you can start the cluster or the application server.

Run the following command:

- AIX `was_profile_root/bin/startNode.sh`
- Linux `was_profile_root/bin/startNode.sh`

Where:

The *was_profile_root* path contains the name of a federated node profile (for example, Custom01).

Starting a cluster

About this task

When you start a cluster, all application servers on the cluster are started.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.
      Where:
      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.
      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Start the cluster:
   a. In the Integrated Solutions Console, click **Servers** → **Clusters** → **WebSphere application server clusters**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Clusters**.

b. Select the check box associated with the name of the cluster.
c. Click **Start**.

Starting a server (console)

About this task

Applications that were running when the server was stopped are restarted automatically.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Start the application server:
   a. In the Integrated Solutions Console, click **Servers** → **Server Types** → **WebSphere application servers**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
   b. Select the check box associated with the name of the server.
   c. Click **Start**.

Starting a server (command line)

Run the following command:

AIX `was_profile_root/bin/startServer.sh server_name -username user_name -password password`

Linux `was_profile_root/bin/startServer.sh server_name -username user_name -password password`

Where:

The *was_profile_root* path contains the name of the application server profile (for example, AppSrv01).

server_name is name of the application server.

user_name represents your WebSphere Application Server administrator user ID.

password represents the password associated with your *user_name*.

Monitoring system performance using WebSphere PMI

WebSphere Performance Monitoring Infrastructure (PMI) provides data that you can use to monitor and tune your application server's performance. PMI consists of several metrics (indicators and counters), each of which provides statistics about a specific aspect of the system's performance. You can enable and disable the metrics according to your specific needs.

The metrics used by WebSphere PMI are sometimes referred to as *key performance indicators*.

For more information about using WebSphere PMI, see the topic *Performance Monitoring Infrastructure (PMI)* in the WebSphere Application Server 7.0 Information Center.

Enabling performance monitoring

Use the Integrated Solutions Console to enable performance monitoring for your application server.

About this task

Complete the following steps to enable PMI-based performance monitoring.

For a full list of supported metrics, refer to the topic *Performance metrics*.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. In the navigation pane, click **Monitoring and Tuning → Performance Monitoring Infrastructure (PMI)**.
3. Select the application server instance.
4. In the Configuration tab, under General Properties, select the check box labeled **Enable Performance Monitoring Infrastructure (PMI)**.
5. Under Currently monitored statistic set, select one of the following:

   None Do not enable any PMI-based metrics.

   Basic Enable a basic set of metrics. (Expand the node to display a list.)

   Extended
   Enable a more comprehensive set of metrics. (Expand the node to display a list.)

   All Enable all of the metrics. (Expand the node to display a list.)

Custom

Enable a set of metrics that you specify. After selecting the radio button, click **Custom** and use the dialog to specify the list of metrics you want.

6. Click **OK**.
7. Click **Save** to save changes to the master configuration.
8. In a clustered configuration, repeat steps 3 on page 119 through 7 for each application server.
9. Restart the server.

Disabling performance monitoring

For best system performance, it is a good practice to disable PMI metrics when the data is no longer needed.

About this task

Complete the following steps to disable PMI-based performance metrics that you no longer need.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.

   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. In the navigation pane, click **Monitoring and Tuning → Performance Monitoring Infrastructure (PMI)**.
3. Select the application server instance. For a clustered configuration, select all of the application server instances.
4. In the Configuration tab, under General Properties, deselect the check box labeled **Enable Performance Monitoring Infrastructure (PMI)**.
5. Click **OK**.
6. Click **Save** to save changes to the master configuration.
7. Restart the server.

Performance metrics

WebSphere Performance Monitoring Infrastructure (PMI) indicators provide a comprehensive set of data to help explain the behavior of applications and the resources they consume.

As an alternative to using the Integrated Solutions Console, you can enable and disable PMI using the `PMI` tag in the Presence Server configuration file, `SystemConfiguration.xml`. See the topic *Configuring performance* for details.

The following tables list the metrics (indicators and counters) that are supported for the Presence Server product.

Rate counters

The following tables list PMI-based indicators used by Presence Server. Use the `PMI` tag in the `SystemConfiguration.xml` file to activate the indicators and specify how you want to monitor their output.

The following PMI-based indicators track the rates at which various events occur.

Table 17. Key performance indicators used in Presence Server: rate counters

Name	Description
SIP providers SUBSCRIBE rate	Rate counter for outgoing subscription requests from Presence Server to SIP providers
Un-PUBLISH rate	Rate counter for incoming un-publish requests
Modify PUBLISH rate	Rate counter for incoming modify publish requests
Initial PUBLISH rate	Rate counter for incoming initial publish requests
URI list un-SUBSCRIBE rate	Rate counter for incoming un-subscribe requests on URI lists
URI list poll-SUBSCRIBE rate	Rate counter for incoming polling subscribe requests on URI lists
URI list re-SUBSCRIBE rate	Rate counter for incoming re-subscribe requests on URI lists
URI list new SUBSCRIBE rate	Rate counter for incoming initial subscribe requests on URI lists
Group poll-SUBSCRIBE rate	Rate counter for incoming polling subscribe requests on groups
Group un-SUBSCRIBE rate	Rate counter for incoming un-subscribe requests on groups
Group re-SUBSCRIBE rate	Rate counter for incoming re-subscribe requests on groups
Group new SUBSCRIBE rate	Rate counter for incoming initial subscribe requests on groups
Single poll-SUBSCRIBE rate	Rate counter for incoming polling subscribe requests on single users
Single un-SUBSCRIBE rate	Rate counter for incoming un-subscribe requests on single users
Single re-SUBSCRIBE rate	Rate counter for incoming re-subscribe requests on single users
Single new SUBSCRIBE rate	Rate counter for incoming initial subscribe requests on single users
Re-PUBLISH rate	Rate counter for incoming re-publish requests
Group NOTIFY rate	Rate counter for outgoing notify requests on groups

Table 17. Key performance indicators used in Presence Server: rate counters (continued)

Name	Description
Single NOTIFY rate	Rate counter for outgoing notify requests on single users
URI list NOTIFY rate	Rate counter for outgoing notify requests on URI lists
Successful SUBSCRIBE response rate	Rate counter for successful responses to incoming subscribe requests
Successful PUBLISH response rate	Rate counter for successful responses to incoming publish requests
Successful NOTIFY response rate	Rate counter for successful responses to incoming notify requests
Failed SUBSCRIBE response rate	Rate counter for failed responses to incoming subscribe requests
Failed PUBLISH response rate	Rate counter for failed responses to incoming publish requests
Failed NOTIFY response rate	Rate counter for failed responses to incoming notify requests

Per second counters

The following PMI-based indicators track the rates per second at which events occur.

Table 18. Key performance indicators used in Presence Server: rates per second

Name	Description
SIP providers SUBSCRIBE per second	Number of outgoing subscription requests per second from Presence Server to SIP providers
Un-PUBLISH per second	Number of incoming un-publish requests per second
Modify PUBLISH per second	Number of incoming modify publish requests per second
Initial PUBLISH per second	Number of incoming initial publish requests per second
URI list un-SUBSCRIBE per second	Number of incoming un-subscribe requests on URI lists per second
URI list poll-SUBSCRIBE per second	Number of incoming polling subscribe requests on URI lists per second
URI list re-SUBSCRIBE per second	Number of incoming re-subscribe requests on URI lists per second
URI list new SUBSCRIBE per second	Number of incoming initial subscribe requests on URI lists per second
Group poll-SUBSCRIBE per second	Number of incoming polling subscribe requests on groups per second
Group un-SUBSCRIBE per second	Number of incoming un-subscribe requests on groups per second
Group re-SUBSCRIBE per second	Number of incoming re-subscribe requests on groups per second

Table 18. Key performance indicators used in Presence Server: rates per second (continued)

Name	Description
Group new SUBSCRIBE per second	Number of incoming initial subscribe requests on groups per second
Single poll-SUBSCRIBE per second	Number of incoming polling subscribe requests on single users per second
Single un-SUBSCRIBE per second	Number of incoming un-subscribe requests on single users per second
Single re-SUBSCRIBE per second	Number of incoming re-subscribe requests on single users per second
Single new SUBSCRIBE per second	Number of incoming initial subscribe requests on single users per second
Re-PUBLISH per second	Number of incoming re-publish requests per second
Group NOTIFY per second	Number of notify responses on groups per second
Single NOTIFY per second	Number of notify responses on single users per second
URI list NOTIFY per second	Number of notify responses on URI lists per second
Successful SUBSCRIBE response per second	Number of successful subscribe responses per second
Successful PUBLISH response per second	Number of successful publish responses per second
Successful NOTIFY response per second	Number of successful notify responses per second
Failed SUBSCRIBE response per second	Number of failed subscribe responses per second
Failed PUBLISH response per second	Number of failed publish responses per second
Failed NOTIFY response per second	Number of failed notify responses per second

Current counters

The following PMI-based indicators track the current state.

Table 19. Current performance counters

Name	Description
Current Group SUBSCRIBE objects	Current number of incoming subscription objects for groups
Current URI lists SUBSCRIBE objects	Current number of incoming subscription objects for URI lists Current outgoing SIP providers
SUBSCRIBE objects	Current number of outgoing SIP providers subscription objects
Current single SUBSCRIBE objects	Current number of incoming single subscription objects

Total counters

The following PMI-based indicators track the total number of times events occur.

Table 20. Key performance indicators used in Presence Server: totals

Name	Description
Total document cache hits	Total number of times a copy of a changed document was saved in cache
Total document cache misses	Total number of times a document changed but was not saved in cache
Total failed NOTIFY responses	Total number of failed responses on incoming notify requests
Total failed PUBLISH responses	Total number of failed responses to incoming publish requests
Total failed SUBSCRIBE responses	Total number of failed responses to incoming subscribe requests
Total incoming SIP PUBLISH messages	Total number of incoming SIP publish messages
Total incoming SIP SUBSCRIBE messages	Total number of incoming SIP subscribe messages
Total sent SIP NOTIFY messages	Total number of SIP NOTIFY messages that were sent
Total document cache hits	Total number of times a copy of a changed document was saved in cache
Total document cache misses	Total number of times a document changed but was not saved in cache
Total failed NOTIFY responses	Total number of failed responses on incoming notify requests
Total failed PUBLISH responses	Total number of failed responses to incoming publish requests
Total failed SUBSCRIBE responses	Total number of failed responses to incoming subscribe requests
Total incoming SIP PUBLISH messages	Total number of incoming SIP publish messages
Total incoming SIP SUBSCRIBE messages	Total number of incoming SIP subscribe messages
Total sent SIP NOTIFY messages	Total number of SIP NOTIFY messages that were sent

Modifying logging

Use the Integrated Solutions Console to specify how data is logged, where the log data is stored, and the output format to use for log data.

About this task

You can modify the general properties of each log, which specifies the output type or location of the log. Use the following steps to adjust the properties for each log type:

1. Log in to the Integrated Solutions Console:

a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

   Where:

   host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

   port is the secured port used to access the console. The default port is *9043*.

   Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.

b. Enter an administrator user ID and password.

c. Click **Log in**.

2. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.

3. Click the name of the server you want to manage.

4. Under Troubleshooting, click **Logging and Tracing**.

5. Click one of the log types. Then click the Configuration tab to make a static change to the system log configuration, or click the Runtime tab to change the configuration dynamically.

   Note: Separate logs for each log type exist for all Java virtual machines (JVMs) on a node, including all application servers and their node agent, if present, as well as for a deployment manager in its own logs directory.

   Here is a list of the available log types:

Option	Description
Diagnostic Trace	Provides information in the `trace.log` about how the WebSphere Application Server components run.
JVM Logs	Used to view and modify the settings for the Java Virtual Machine (JVM). The `System.out` log (`SystemOut.log`) is used to monitor the health of the WebSphere Application Server. The `System.err` log (`SystemErr.log`) contains exception stack trace information used to perform problem analysis.
Process Logs	Created when redirecting the standard out and standard error streams of a process to independent log files, the `native_stdout.log` and `native_stderr.log`, respectively.
IBM Service Logs	Also known as the activity log. Records the WebSphere Application Server messages that are written to the `System.out` stream and special messages that contain extended service information that you can use to analyze problems.

Option	Description
Change Log Detail Levels	Controls which events are processed by Java logging, by using log levels. You can assign logging levels to individual trace loggers or to trace groups. (Trace loggers and groups are listed in the topic *Trace loggers*.) All Presence Server logs are located in `com.ibm.presence`. To view them, you can change the package log level so that the logs will appear in the trace file, for example `com.ibm.presence.*=all` or `com.ibm.presence.*=fine`.

6. When you are finished making your changes, click **Apply**.
7. Click **OK**.
8. Click **Save** to save changes to the master configuration.
9. Optional: If you made a static change to the configuration, restart the application for your changes to take effect.

Using IBM Tivoli License Manager

The IBM Tivoli® License Manager (ITLM) product is used to detect where IBM products are both installed and running. ITLM is installed with each component of the IBM IMS solution.

Compatibility

This release of IMS runs with ITLM server version 2.2.

Note: The ITLM agent version 2.2 might not be compatible with all versions of the operating systems supported by IBM WebSphere Presence Server Component. Review the ITLM documentation carefully to determine which operating systems, maintenance levels, and Linux kernel versions are supported.

Installation

During the installation of Presence Server, the inventory signatures for ITLM are installed in the *was_profile_root*`/installedApps` directory.

Inventory Signature

The ITLM inventory signature file uniquely identifies the product and is installed with the Presence Server EAR into WebSphere Application Server. It is located within the EAR file in the `TIVREADY` subdirectory, and its name is `IMSPRE0603.SYS2`.

License file

Presence Server comes with a license file that is not associated with ITLM enablement, but is still important for licensing purposes. This text file specifies the customer's entitlement of installation and use of the product.

Usage Signature

The ITLM usage signature, generated by the Software Catalogue Signature team, is used to identify each IMS component as a J2EE product.

Chapter 6. Troubleshooting IBM WebSphere Presence Server Component

Logs store information to help you troubleshoot problems installing, configuring, and using IBM WebSphere Presence Server Component.

Using ISA 4.0 add-ons to communicate with IBM Support

To help you communicate with IBM Support, IBM Support Assistant (ISA) 4.0 product add-ons are available on the Web for IBM WebSphere Presence Server. You can install the add-ons for selected products and features using the ISA graphical user interface.

About this task

You can open electronic service requests using the ISA add-ons. If you want to send log files associated with the service request, you must install and use the add-on for the version of WebSphere Application Server that you are running. It collects logs, trace files, and configuration information to send to IBM Support.

To install the product add-ons, perform the following steps:

1. Download and install ISA, using the instructions found on the IBM Support Assistant Web site.
2. Launch the IBM Support Assistant Workbench.
3. Click **Update** → **Find new** → **Product Add-ons**.
4. In the Product Add-ons window, select the ISA product add-ons you want to install. The add-ons are categorized by product family.
   a. Expand the **WebSphere** product family.
   b. Check one or more products for which you want to install add-ons.
   c. Click **Next**.
5. In the Tools Add-ons window, select any additional ISA add-ons you want to install. Then click **Next**.
6. Review the license information for the add-ons you have selected, and click **I accept the terms in the license agreements**.
7. Click **Next**.
8. Click **Finish**.
9. Restart the IBM Support Assistant when the installation has completed.

Troubleshooting the database script

You can take a number of different actions if errors occur when you run the database preparation scripts .

Problem	Solution
The script fails with an `access denied` error.	Add execute permission for the script, for example: DB2 `chmod 777 Scriptname.sh` Oracle `chmod 777 Scriptname.sh`
The script fails with a `file or directory does not exist` error.	Run the `dos2unix` command, for example: DB2 `dos2unix Scriptname.sh` Oracle `dos2unix Scriptname.sh`
The attempt to unpack `IBMPresenceServerDbPackage_7.0.tar` fails with an `access denied` error.	Add execute permission for the `.tar` file, for example: DB2 `chmod 777 IBMWebSpherePresenceServer.tar`
You cannot connect to the database.	Verify that the DB2 database manager or the Oracle server is started. DB2 Open the script using a text editor and verify the following: • The correct values are specified for DBUSER and DBUSERPW • DBCREATE is set to TRUE (for new databases) • DBPORT is set to the port that is being used by your DB2 server (the default is 50000) • Linux DBSERVER matches the Linux `hostname` command value, if you are running the script directly on the database server Oracle Shut down and then restart the database. The easiest way to do this is through the Oracle console.

Monitoring log messages

IBM WebSphere Presence Server Component can write system messages to several general purpose logs. Logging provides information about important lifecycle events, warnings, and errors that should be addressed by an administrator.

By default, IBM WebSphere Presence Server Component logs its informational messages to the WebSphere Application Server JVM log (`SystemOut.log`) and its trace messages to the WebSphere Application Server trace log (`trace.log`). IBM WebSphere Presence Server Component stores all exceptions to the `SystemErr.log`. Both log files are located in the `logs` directory:

- AIX *was_profile_root*`/logs/`*server_name*
- Linux *was_profile_root*`/logs/`*server_name*

Note: *was_profile_root* is the directory for a WebSphere Application Server Network Deployment profile called *profile_name*. By default, this directory is:

- AIX `/usr/IBM/WebSphere/AppServer/profiles/profile_name`
- Linux `/opt/IBM/WebSphere/AppServer/profiles/profile_name`

In a standalone environment, *profile_name* is the name of the application server profile. In a clustered environment, *profile_name* is the name of a federated node profile.

Each error, warning, or informational log message should include a message code which is used to identify the message. Additionally, each message can be identified by the date, timestamp, thread number, and severity. For example:

[4/17/09 17:32:42:086 EDT] 00000021 StorageEngine I CWSPS0101I: Looking up Data source for java:comp/env/DBAlias alias name

Comprehensive information about working with message logs may be found in the WebSphere Application Server Network Deployment information center.

Viewing and modifying logs

Use the Integrated Solutions Console to specify how data is logged, where the log data is stored, and the output format to use for log data.

About this task

You can modify the general properties of each log, which specifies the output type or location of the log. Use the following steps to adjust the properties for each log type:

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have "http" instead of "https" in the URL.

   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
3. Click the name of the server you want to manage.
4. Under Troubleshooting, click **Logging and Tracing**.
5. Click one of the log types. Then click the Configuration tab to make a static change to the system log configuration, or click the Runtime tab to change the configuration dynamically.

Note: Separate logs for each log type exist for all Java virtual machines (JVMs) on a node, including all application servers and their node agent, if present, as well as for a deployment manager in its own logs directory.

Here is a list of the available log types:

Option	Description
Diagnostic Trace	Provides information in the `trace.log` about how the WebSphere Application Server components run.
JVM Logs	Used to view and modify the settings for the Java Virtual Machine (JVM). The `System.out` log (`SystemOut.log`) is used to monitor the health of the WebSphere Application Server. The `System.err` log (`SystemErr.log`) contains exception stack trace information used to perform problem analysis.
Process Logs	Created when redirecting the standard out and standard error streams of a process to independent log files, the `native_stdout.log` and `native_stderr.log`, respectively.
IBM Service Logs	Also known as the activity log. Records the WebSphere Application Server messages that are written to the `System.out` stream and special messages that contain extended service information that you can use to analyze problems.
Change Log Detail Levels	Controls which events are processed by Java logging, by using log levels. You can assign logging levels to individual trace loggers or to trace groups. (Trace loggers and groups are listed in the topic *Trace loggers*.) All Presence Server logs are located in `com.ibm.presence`. To view them, you can change the package log level so that the logs will appear in the trace file, for example `com.ibm.presence.*=all` or `com.ibm.presence.*=fine`.

6. When you are finished making your changes, click **Apply**.
7. Click **OK**.
8. Click **Save** to save changes to the master configuration.
9. Optional: If you made a static change to the configuration, restart the application for your changes to take effect.

Results

After your configuration changes take effect, you will be able to view log data in the locations, and in the output formats, that you have specified. Note that certain logging settings can affect the performance of your system.

Enabling trace

Trace logs show trace events such as function entries and exits, component events, and debugging activities. Use the administration console to enable trace for a process.

About this task

You can configure the IBM WebSphere Presence Server Component to start in a trace-enabled state by setting the appropriate configuration properties.

You can control how much detail each logger records by adjusting the log level details. Because the loggers are grouped hierarchically, setting the trace level on one logger also sets all subsequent loggers to the same level. Altering the tracing levels impact the performance of the system.

Enable and configure trace by completing the following steps:

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.
   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. In the navigation pane, click **Servers** → **Server Types** → **WebSphere application servers**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
3. Click the name of the server you want to manage.
4. Click **Troubleshooting** → **Logging and Tracing**.
5. Click **Diagnostic Trace Service**.
6. Configure your trace options:
   a. Display the Runtime tab.
   b. To disable tracing, select **File** and then select **None**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, disable tracing by selecting **Enable log** and then substituting **disabled** in place of **enabled**.
   c. Click **Change Log Level Details**.
   d. Click **Components** to view all loggers for the individual components.
   e. Click **+** to show the *children* of the logger.
   f. Click ***logger_name*** to change the log details. To enable tracing on specific components of IBM WebSphere Presence Server Component, use `com.ibm.presence.*=all` as a logger group name.
   g. Choose the appropriate level of tracing.

      Remember: When you change the level for a logger, the change is propagated to the children of the logger.

      For additional information regarding trace levels, click **?** in the title bar of the panel to open the help page.

7. Click **OK**.
8. Click **Save**.

Results

The specified traces are enabled for the current server session. To make the changes permanent, use the Configuration tab rather than the Runtime tab when you configure the trace options. Note that when you use the Configuration tab, you will need to restart the server for your changes to take effect.

Selecting trace loggers

The level of tracing is determined by the log level details you select for the loggers. Loggers are organized hierarchically. The children of the logger will inherit the parent log level by default, but it can be changed by defining the level of tracing on each specific logger.

To control the trace level for the Trust Association Interceptor, use the options on the `com.ibm.imsconnector.*` trace group. For more specific levels of tracing, use the following trace groups, which are relevant to the Trust Association Interceptor:

Table 21. TAI trace groups and trace loggers

Trace group	Trace loggers
com.ibm.imsconnector.tai.*	*BaseIMSInterceptor *HttpInterceptor *SipInterceptor

To control the trace level for the Presence Server component and its subcomponents, use the options on the `com.ibm.presence.*` trace group. For more specific levels of tracing, use the following trace groups, which are relevant to Presence Server:

Table 22. Presence trace groups and trace loggers

Trace group	Trace loggers
com.ibm.presence.*	*aggregators.* *authorization.* *channel.* *charging.* *component.* *configuration.* *contentindirection.* *document.* *externalsource.* *init.* *management.* *managing.* *monitor.* *pmi.* *provider.* *rls.* *service.* *services.* *sharedlists.* *siplets.* *storage.* *subscribe.* *timers.* *util.* *winfo.* *xdms.* *xml.*

AG attachments:

- com.ibm.websphere.sca.soap.attachments.*
- com.ibm.ws.sca.soap.attachments.*

AG handlers:

- com.ibm.soa.esb.global.handlers.*
- com.ibm.sca.connections.handlers.*
- com.ibm.ws.sca.soap.attachments.handlers.*

Tracing a SIP container

To control the trace level for the SIP container, use the options on the `com.ibm.ws.sip.*` trace group. For more detailed information about tracing a SIP container or CEI, refer the WebSphere Application Server Network Deployment Information Center.

Messages

A message explains a problem and suggests a user action. In addition, each message ID includes a component ID, a number, and a letter that indicates the type of message: Informational, warning, or error.

Message key

Each sub component has a unique message identifier to help you determine the origin of the message.

Standard format

The standard message format is: *AAAANNNNS*

- *AAAA* represents the component identifier (typically four or five characters).
- *NNNN* represents a four digit identifier.
- *S* represents the type of messages. There are three message types:
  - I represents informational messages.
  - W represents warning messages.
  - E represents error messages.

IBM WebSphere Presence Server Component messages

The following message identifier is used for Presence Server:

Table 23. Presence Server

Component identifier	Component description
CWSPS	Presence Server messages including the handling of PUBLISH, SUBSCRIBE, and NOTIFY requests; integration with SIP external sources; integration with an XDMS server, database, and JNDI; JMS errors and warnings.

Trust Association Interceptor messages

The following message identifier is used for Trust Association Interceptor:

Table 24. Trust Association Interceptor

Component identifier	Component description
DHAT	TAI messages

Chapter 7. Developing applications and using product features

IBM WebSphere Presence Server Component provides an API for developing customized authorization policies.

Extending Presence Server authorization

IBM WebSphere Presence Server Component provides APIs with which you can develop customized authorization policies.

Developers can use the authorization APIs to write pluggable applications that provide authorization information. This information either allows or disallows subscription to a presentity.

If you implement a custom authorization application, you must:

- Add the `IBMWebSpherePresenceServerAPI.jar` file to the class path. Refer to the topic *Adding the API jar file to the class path*.
- Enable the `externalAuthorization` tag and disable the `presenceRules` tag in the `SystemConfiguration.xml` file. Refer to the topic *Configuring authorization and authentication*.
- Disable white and black lists in the `SystemConfiguration.xml` file using the `authorizationLists` tag. Refer to the topic *Configuring authorization and authentication*.

General information about authorization

Presence Server can be configured to work with only one authorization service at a time. Consider the following factors when setting up customized authorization policies.

After it registers an authorization service, Presence Server will reject any other registrations to the *AuthorizationService Manager*. When it receives a new subscription, Presence Server will use the authorization service to determine if the subscription is allowed, and one of the following happens:

- If the subscription is allowed, then the subscriber will receive the presence information for the presentity to which it is subscribed.
- If the subscription is politely blocked, then the user attempting the subscription will receive an empty presence document. The subscription will stay active.
- If the subscription is confirmed, then the user attempting the subscription will receive an empty presence document. The subscription will stay pending.
- If the subscription is blocked, then the user attempting the subscription will receive an empty presence document. The subscription will be terminated.

Proactive authorization

A proactive authorization is one where the authorization policies have been defined to identify the authorized and unauthorized users to subscribe for a given presence data. Presence Server consults with the stored permission policies, using the authorization API, to determine if the requester, or watcher, is authorized to subscribe on a given presentity. If the watcher is unauthorized to subscribe,

Presence Server will notify the watcher with empty presence data. Then, Presence Server indicates that the subscribed presentity is unavailable independently on current presence information.

Reactive authorization

The combination of watcher information and presence authorization rules enables the use of reactive authorization, where authorization occurs through direct user intervention. A user can subscribe to the watcher information for his or her presentity and thus find out when a new watcher is added who is not covered by the existing authorization rules. The user may then add a new authorization rule for the new watcher.

How Presence Server uses the authorization service

IBM WebSphere Presence Server Component allows an authorization service registration using **AuthorizationServiceManager.registerAuthorizationService**.

For each new SUBSCRIBE request, Presence Server invokes **AuthorizationService.doSubscribeIsAllowedUser**. **AuthorizationService.doSubscribeIsAllowedUser** subscribes a user to the permission notifications for the subscription.

When the subscription expires, IBM WebSphere Presence Server Component uses **AuthorizationService.doUnsubscribeIsAllowedUser** to stop receiving notifications on permission policies.

While the subscription to the authorization service is active, IBM WebSphere Presence Server Component will receive notifications of the subscription permissions. IBM WebSphere Presence Server Component examines the notifications returned by **AuthorizationRulesListener.onSubscriptionIsAllowed**.

Using the data in the **AuthorizationRules** object, **AuthorizationRulesListener.onSubscriptionIsAllowed** retrieves an authorization flag that indicates the subscription permission. If the subscription is allowed, IBM WebSphere Presence Server Component gets the current presence information from the database and sends it to the subscriber. If the subscription is not allowed, IBM WebSphere Presence Server Component sends an empty NOTIFY response.

Error codes

The *AuthorizationErrorCodes* class defines errors to be sent from Presence Server to an authorization service on a failed registration. As long as the subscription is active, Presence Server continues and receives updates of any changes in the permission policy, and thus reflect these changes in the existing subscriptions.

Developing an authorization application

To develop an authorization application you need to understand how the authorizationService interface passes information.

Restrictions

The following restrictions apply when you implement the Authorization service:

- Only one authorization service registration is allowed.

- The Authorization service should be deployed on the same WebSphere Application Server server as Presence Server.
- Each new SUBSCRIBE request will use the following method to subscribe to permission notifications for that subscription: `AuthorizationService.doSubscribeIsAllowedUser()`.
- Presence Server will use the same method–`AuthorizationService.doUnsubscribeIsAllowedUser()`–to stop receiving notifications on permission policies when the subscription is no longer active. As long as the subscription to the Authorization service is active, it will send notification to Presence Server.

Note: For communication to take place with the external authorization service, the Presence Server `externalAuthorization` tag must be enabled (the `enable` value set to true). See the topic *Configuring authorization* for more information.

Implementing the Authorization service

You can implement the Authorization service by the following steps:

1. Create an EJB project and implement the AuthorizationService interface:

```
public interface AuthorizationService {
public void doSubscribeIsAllowedUser (AuthorizationData authorization Data,
  AuthorizationRulesListener listener) throws RemoteException;
public void doUnsubscribeIsAllowedUser (AuthorizationData authorization Data)
  throws RemoteException;
 }
```

2. Register with the Presence Server using:

```
AuthorizationServiceManager.registerAuthorizationService (AuthorizationData data) API.
```

   This mechanism is used in place of a JNDI mechanism. When registering to Presence Server a reference to an AuthorizationService EJB, which is encapsulated within AuthorizationData, is passed to Presence Server.

3. Presence Server calls method doSubscribeIsAllowedUser( ) for every new subscription that it receives. If the subscription is on a resource-list, this method is called for each member in the resource-list. This method is implemented by a third party and passes these two objects to the AuthorizationService EJB.
   - **AuthorizationData**: This object encapsulates the information about *To* or *From* headers that the AuthorizationService uses to retrieve authorization. The logic of this object needs to be implemented by a third party.
   - **AuthorizationRulesListener**: This object provides a handle to a listener in Presence Server where authorizationService uses the method

     ```
     onSubscriptionIsAllowed(AuthorizationData authorizationData data, AuthorizationRules Rule)
     ```

     This method can be used synchronously or asynchronously to notify Presence Server for authorization results or changes. The third party will instantiate the object *rule*, which contains the authorization result in the form of an integer representing one of the following actions: `Block`, `Confirm`, `Polite_Block`, or `Allow`.

     Presence Server will then examine the *rule* object to determine whether or not it can retrieve the current presence information for that presence entity, and will then send it to the subscriber.

     Note that the `Polite Block` action was called `Block` in previous versions of Presence Server.

Adding the API JAR file to the class path

To make the Authorization Service interface accessible to external applications, you need to add the `IBMWebSpherePresenceServerAPI.jar` file to the class path.

Before you begin

To complete this procedure, you will need the file IBMWebSpherePresenceServerAPI.jar . This file is provided with theIBM WebSphere Telecom Toolkit.

About this task

Follow these steps to add IBMWebSpherePresenceServerAPI.jar to the class path.

1. Log in to the Integrated Solutions Console:
   a. Open a browser and navigate to the following URL: https://*host_name*:*port*/ibm/console.

      Where:

      host_name is the fully qualified host name of the server where the application or the network deployment manager is deployed.

      port is the secured port used to access the console. The default port is *9043*.

      Note: The default unsecured port is *9060*. If you use 9060, you must have ″http″ instead of ″https″ in the URL.

   b. Enter an administrator user ID and password.
   c. Click **Log in**.
2. Click **Servers** → **Server Types** → **WebSphere application servers**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Servers** → **Application servers**.
3. Click *server_name*. Where *server_name* is the application server on which IBM WebSphere Presence Server Component is deployed.
4. Under Server Infrastructure, click **Java and Process Management**.
5. Click **Process Definition**.
6. Under Additional Properties, click **Java Virtual Machine**.
7. Under Additional Properties, click **Custom Properties**.
8. Click **New**.
9. In the **Name** field type `ws.ext.dirs`.
10. In the **Value** field type the path to the file IBMWebSpherePresenceServerAPI.jar .
11. Click **Apply**.
12. Click **Save** to save changes to the master configuration.
13. Restart the application server.

Using the REST API sample

The REST API sample is a J2EE Web application that can be used as a Web client for accessing the Presence Server REST API.

Description of the REST API sample

You can use the REST API sample to publish, modify, and delete information for a presentity. You can also use it to fetch information about a presentity or a list of presentities.

The sample includes:

- JavaScript code for parsing lists of multipart/related documents into JavaScript objects
- Example code that uses the parser for displaying lists that are formatted

The following files are included in the sample:

- `ReadMe.txt`: Description of the sample.
- `listParser.js`: JavaScript code for parsing lists of multipart/related documents into JavaScript objects.
- `messages.properties`: File containing all of the messages in the sample, used for localization. When adding a new locale, simply copy this file with the correct locale name and change the messages values.
- `pub.jsp`: The main Publish page, used to publish, modify, or delete information for a presentity.
- `sub.jsp`: The main Fetch page, used for fetching information about a presentity or a list of presentities.
- `subFormated.jsp`: An example of parsing lists information into JavaScript objects. The page uses the JavaScript parser for displaying lists that are formatted.

Installing and configuring the REST API sample

Before using the REST API sample, install the application and perform initial configuration.

1. Update the Presence Server configuration to enable support for the REST API. For more information, refer to the topic *Configuring REST* in the Presence Server information center.
2. Install the REST API sample application EAR file as a J2EE application on WebSphere Application Server.
3. Configure the REST API sample application:
   a. Launch the WebSphere Integrated Solutions Console.
   b. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   c. Click **Presence HTTP Gateway**.
   d. Click **Environment entries for Web modules**.
   e. Set the `restAddress` parameter value to the full URL of the Presence Server Rest API. Example: `http://presence.rest.address:9080/presence/`

What to do next

You are ready to run the sample application.

Running the REST API sample

The REST API sample is a J2EE Web application, accessed through a Web browser.

About this task

The default context root is `/gui/`.

To access the sample application, you will need to use a Web browser–either Internet Explorer or Firefox.

1. Open the Web browser and enter the REST sample URL. Example: `http://server.address:9080/gui/`
2. From the actions listed in the right frame, choose whether you want to publish or fetch.
3. In the Publish page you can do the following:
   - Set the publish document.
   - Set the user URI name that will be used for the publisher and published users.
   - Publish the document if this is the first document.
   - Modify the document if you have already published information.
   - Delete the document if you have already published information.
   - View the Status of the last request, for example the return code and the Etag number.
4. In the Fetch page you can do the following:
   - Send a fetch request for a single user.
   - Send a fetch request for a list.
   - View the fetch response (the user/list information) in the main text box.
   - View a formatted list of information by entering the list URI and clicking the **Formatted** button.
     – In the Formatted List page, you can view the list structure.
     – If one of the list resources has information (pidf information), you can select it and then view the information in the lower text box.

Using the SIP sample

The SIP sample is a J2EE Web application that can be used as a SIP client. Working in conjunction with the REST sample, it can send SIP requests to Presence Server and receive SIP responses from Presence Server.

Description of the SIP sample

The sample application is written on top of the SIP container. Client applications can create SIP messages by sending HTTP requests to the sample. The SIP responses are translated back to HTTP responses and sent to the clients.

The following operations are supported:

- Publish, modify, or delete information for a presentity
- Fetch information about a presentity

You can use the REST API sample to send HTTP requests to the SIP sample. (Refer to the topic *Using the REST API sample* for more information.)

This sample is also a sample of a converge application. Because SIP sessions and HTTP sessions can share the same SIP application sessions, data sharing can take place between SIP and HTTP applications. This sample application utilizes data sharing on the SIP application session in order to synchronize Siplets and Servlets.

Installing and configuring the SIP sample

Before using the SIP sample, install the application and perform initial configuration.

1. Install the SIP sample application EAR file as a J2EE application on WebSphere Application Server.
2. Configure the SIP sample application:
   a. Launch the WebSphere Integrated Solutions Console.
   b. Click **Applications** → **Application types** → **WebSphere Enterprise Applications**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Applications** → **Enterprise Applications**.
   c. Click **Presence SIP Sample**.
   d. Click **Environment entries for Web modules**.
   e. Set the `PresenceServerAddress` parameter value to the SIP address for the Presence Server instance. Example: `sip:localhost:5060;transport=TCP`
   f. Click **OK**.

What to do next

You are ready to run the sample application.

Running the SIP sample

The SIP sample is a J2EE Web application, accessed through a Web browser.

About this task

The default context root is `/sample/`.

To access the sample application, you will need to use a Web browser–either Internet Explorer or Firefox.

1. Install the REST sample application and configure it to send HTTP requests to the SIP sample:
   a. Follow the instructions in the topic *Installing and configuring the REST API sample* to install the sample.
   b. Set the value of the **restAddress** configuration parameter to the full URI for the SIP sample. Example: `http://sip.sample.server.address:9080/sample/`
2. Open the Web browser and enter the REST sample URL. Example: `http://server.address:9080/gui/`
3. From the actions listed in the right frame, choose whether you want to publish or fetch.
4. In the Publish page you can do the following:
   - Set the publish document.
   - Set the user URI name that will be used for the publisher and published users.
   - Publish the document if this is the first document.
   - Modify the document if you have already published information.
   - Delete the document if you have already published information.
   - View the Status of the last request, for example the return code and the Etag number.

5. In the Fetch page you can do the following:
   - Send a fetch request for a single user.
   - View the fetch response (the user information) in the main text box.

   Note: The SIP sample does not support subscriptions on lists. You cannot, therefore, select the option of sending a fetch request for a list.

Starting a custom REST project

The first step toward deploying a REST interface is to set up a custom REST project in the Rational® Application Developer workspace.

About this task

Follow these instructions to set up your custom REST project.

1. Launch Rational Application Developer with a new workspace.
2. Import the sample or samples needed for the project.
   a. Click **Help** → **Samples** → **Technology samples** → **Telecom Enablement Feature** → **Telecom Enablement Samples** → **Presence**
   b. Follow the instructions on the Setup page of the Samples Gallery to configure presence for use with external sources.
   c. Click **Help** → **Samples** → **Technology samples** → **Telecom Enablement Feature** → **Telecom Enablement Samples** → **Presence**
   d. Click **Import Presence REST**

      Note: Select the sample for use with WebSphere Application Server 7.0.0.1 or 6.1.0.21, depending on your server configuration.
   e. Click **Finish**.
3. Use the sample `presence.sample.rest.war` to test HTTP fetch and publish requests or customize it to make your own REST interface.
4. Right click **presence.sample.rest.war** and select **Run** → **Run on Server** to launch the sample REST Web interface. Alternatively, use a Web browser to point to the REST Web interface using the address which ends in `/gui`–for example, `http://localhost:9081/gui`
5. Expand the `presence.sample.rest.war` folder to locate the Readme file, which contains further instructions for developing and deploying a REST interface. The Readme file includes configuration instructions for making the REST sample work with the SIP sample.

Migrating your applications from a previous release of IBM WebSphere Presence Server Component

For users of previous versions of the Presence Server product, it is not necessary to modify your existing application programs to make them compatible with the current version.

Existing Presence Server client applications can interact with Presence Server version 7.0. New subscriptions are routed to the version 7.0 instance.

A network device or client that interacts with Presence Server version 6.2 does not need to change or upgrade to interact with Presence Server version 7.0.

Accessing data from SIP external sources

IBM WebSphere Presence Server Component provides ways in which users can gain access to a rich supply of data from external sources, such as S-CSCF and external communities.

Presence Server provides a sample application to demonstrate the behavior of a SIP external source. You can modify the sample application to interact with external sources in your own enterprise.

The sample application is a SIPp script and performs the following functions:

- Accepts incoming subscribe requests, responding with a 200 OK response and a NOTIFY message that contains a simple presence document. The document includes the identity of the subscribed presentity.
- Simulates state changes for the subscribed presentity by periodically sending NOTIFY messages with changes in the presence information (for example, a basic status change).
- Stops sending NOTIFY messages when it receives an unsubscribe request from Presence Server.

The sample application is found in the Telecom Application Enablement Feature in the Samples Gallery.

Note: SIPp is a free, open-source test tool with which you can create applications that generate SIP traffic. For more information about SIPp, refer to the SIPp Web site (http://sipp.sourceforge.net/).

Chapter 8. Reference information

Information about supported standards, directory conventions, and terminology are provided as additional reference information to help you.

Changes to this edition

Since the last edition of this information, the following changes have been made.

Table 25. Change history for the product documentation

Edition	Date	Changes
First Edition	April 2009	First issue of the product documentation.

Documentation conventions

Typographical conventions are used to make the documentation easier to understand.

The following conventions are used throughout the documentation:

- Variables are italicized. Italicized information indicates that you should substitute information from your environment for the value. For example:

  http://*host_name*:*port_number*
- Variables are used to indicate installation directories. The variable links to information with the default paths. For example:

  was_root/logs
- Images are used to indicate information specific to one operating system or database software. For example:

  Linux *was_root*/installableApps/TWSS-Services
- Values that you must type display in `monospace` font.
- User interface elements display as **boldfaced** text.
- Links to related information for each topic are provided at the bottom of the topic.

Directory conventions

References in the documentation are for default directory locations. This topic describes the conventions in use for WebSphere Application Server Network Deployment.

Default product locations when the root user or an administrator user installs the product

The root user or administrator is capable of registering shared products and installing into the default system-owned directories. These file paths are default locations, but you can install the products and create profiles in any directory where you have write access. Multiple installations of any of these products or components require multiple installation locations.

was_root
The following list shows default installation root directories for WebSphere Application Server Network Deployment:

- AIX /usr/IBM/WebSphere/AppServer
- Linux /opt/IBM/WebSphere/AppServer

was_profile_root
The following list shows the default directory for a WebSphere Application Server Network Deployment profile named *profile_name*:

- AIX /usr/IBM/WebSphere/AppServer/profiles/*profile_name*
- Linux /opt/IBM/WebSphere/AppServer/profiles/*profile_name*

installed_apps_root
The following list shows the default directory for installed applications within a profile named *profile_name*:

- AIX /usr/IBM/WebSphere/AppServer/profiles/*profile_name*/installedApps/*cell_name*/
- Linux /opt/IBM/WebSphere/AppServer/profiles/*profile_name*/installedApps/*cell_name*/

db_client_root
The following list shows default installation root directories for the database clients:

- DB2 AIX /usr/IBM/db2/V9.5
- DB2 Linux /opt/IBM/db2/V9.5
- Oracle /home/oracle/app/oracle/product/11.1.0

Glossary

This glossary contains terms that pertain specifically to the IBM WebSphere software for Telecom: IBM WebSphere IP Multimedia Subsystem Connector V6.2, IBM WebSphere Presence Server V7.0, IBM WebSphere Telecom Web Services Server V7.0, and IBM WebSphere XML Document Management Server V7.0.

The glossary also contains relevant terms from the IBM English Terminology Database.

A

Administrative console
A graphical interface that guides the user through systems administration tasks such as deployment, configuration, monitoring, starting and stopping applications, services, and resources.

Application Manager
In Common Desktop Environment (CDE), a window containing objects representing the system actions available to you.

application programming interface (API)
An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

B

C

Call Notification
A Parlay X Web service that notifies Web clients of specific call events established through the SIP protocol for a specific called party. Call Notification supports regular SIP and IMS call flows.

CDMA2000
A set of 3G standards based on earlier 2G CDMA technology.

charge header support vector utility
A utility class that handles Session Initiation Protocol (SIP) messages, for charging interactions.

Charging Collection Function (CCF)
Defined by the 3GPP group as the entity that receives information through Diameter messages pertaining to Charging Data Records.

cluster
A group of servers that are managed together and participate in workload management. See also horizontal cluster, vertical cluster.

code division multiple access (CDMA)
A form of multiplexing where the transmitter encodes the signal using a pseudo-random sequence, which the receiver also knows and can use to decode the received signal. Each different random sequence corresponds to a different communication channel.

common base event
A specification based on XML that defines a mechanism for managing events, such as logging, tracing, management, and business events, in business enterprise applications.

common event infrastructure (CEI)
A core technology of the IBM Autonomic Computing initiative that provides basic event management services, including consolidating and persisting raw events from multiple, heterogeneous sources and distributing those events to event consumers.

D

demilitarized zone (DMZ)
A configuration including multiple firewalls to add layers of protection between a corporate intranet and a public network, like the Internet.

E

Enhanced Data Rate for GSM Evolution (EDGE)
A development of GSM that allows for the faster delivery of advance mobile services such as full multimedia messaging.

Enterprise JavaBeans
A component architecture defined by Sun Microsystems for the development and deployment of object-oriented, distributed, enterprise-level applications.

event state compositor (ESC)
A server that processes PUBLISH requests and is responsible for composing an event state into a complete, composite event state of a resource.

F

frequency division duplex (FDD)
The application of FDMA to separate outbound and returning signals. The uplink and downlink subbands are said to be separated by the ″frequency offset.″

frequency division multiple access (FDMA)
An access technology that is used by radio systems to share the radio spectrum. The terminology “multiple access” implies the sharing of the resource among users, and “frequency division” describes how the sharing is done by allocating users with different carrier frequencies of the radio spectrum.

G

General Packet Radio Service (GPRS)
A mobile data service available to users of GSM mobile telephones. It is often described as ″2.5G.″ that is, a technology between the second (2G) and third (3G) generations of mobile telephony. It provides moderately fast data transfer by using unused TDMA channels in the GSM network.

Global System for Mobile Communications (GSM)
A second-generation (2G) standard for digital cellular telephone systems, which originated in Europe and is now used in countries across the globe. GSM networks use digital signals and narrowband TDMA, in conformance to a standard developed by the 3GPP, to support voice, data, text, and facsimile transmissions. The world’s most popular standard for mobile telephones, GSM service is used by more than 1.5 billion people across more than 210 countries and territories.

Groupe Special Mobile (GSM)
See Global System for Mobile Communications (GSM).

H

home subscriber server (HSS)
The server that manages the database of all subscriber and service data in an IMS network. Parameters include user identity, allocated S-CSCF name, roaming profile, authentication parameters, and service information.

horizontal cluster
A cluster in which the cluster members exist on multiple physical servers, effectively and efficiently distributing the workload of a single instance. Horizontal clustering provides the ability to build in redundancy and failover, to easily add new members to increase capacity, and to improve scalability by adding heterogeneous systems into the cluster. See also vertical cluster.

hypertext transfer protocol (HTTP)
An Internet protocol that is used to transfer and display hypertext and XML documents on the Web. Hypertext Transfer Protocol Secure (HTTPS).

I

IMS Application Server (AS)
Defined by the 3GPP to be the functional component that invokes applications (usually SIP applications) that provide services to IMS users.

Institute of Electrical and Electronics Engineers (IEEE)
A professional society accredited by the American National Standards Institute (ANSI) to issue standards for the electronics industry.

Internet Engineering Task Force (IETF)
The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet. The IETF consists of numerous working groups, each focused on a particular problem. Internet standards are typically developed or reviewed by individual working groups before they can become standards.

IP Multimedia Subsystem (IMS)
A network services architecture defined by 3GPP that enables support for IP multimedia applications based on SIP and IETF Internet protocols. IMS can use a variety of access methods, including wire-line IP, IEEE 802.11, 802.15, CDMA, and packet data transmission systems such as GSM, EDGE, and UMTS.

J

Java 2 Platform, Enterprise Edition (J2EE)
An environment for developing and deploying enterprise applications, defined by Sun Microsystems Inc.

Java API for XML-based RPC (JAX-RPC)
A specification that describes application programming interfaces (APIs) and conventions for building Web services and Web service clients that use remote procedure calls (RPC) and XML. JAX-RPC is also known as JSR 101.

Java authentication authorization service (JAAS)
In J2EE technology, a standard API for performing security-based operations. Through JAAS, services can authenticate and authorize users while enabling the applications to remain independent from underlying technologies.

Java Database Connectivity (JDBC)
An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based and XQuery-based database access.

Java Management Extensions (JMX)
A means of doing management of and through Java technology. Developed by Sun Microsystems, Inc., and other leading companies in the management field, JMX is a universal, open extension of the Java programming language for management that can be deployed across all industries, wherever management is needed.

Java Messaging Service (JMS)
An application programming interface that provides Java language functions for handling messages.

Java Naming and Directory Interface (JNDI)
An extension to the Java platform that provides a standard interface for heterogeneous naming and directory services.

Java virtual machine (JVM)
A software implementation of a central processing unit that runs compiled Java code (applets and applications).

K

L

Lightweight Directory Access Protocol (LDAP)
An open protocol that uses TCP/IP to provide access to directories that support an X.500 model and that does not incur the resource requirements of the more complex X.500 Directory Access Protocol (DAP). For example, LDAP can be used to locate people, organizations, and other resources in an Internet or intranet directory.

location generator
The entity that initially determines or gathers the location of the target and creates location objects that describe the location of the target.

location object
An object that conveys location information (and possibly privacy rules) to which Geopriv security mechanisms and privacy rules are to be applied.

location recipient
The entity that receives location information. It might have asked for this location explicitly (by sending a query to a location server), or it might receive this location asynchronously.

location server
an element that receives publications of Location Objects from Location Generators and may receive subscriptions from Location Recipients. An entity that receives location objects published by a location generator, receives queries from location recipients, and applies privacy rules designed by the rule maker, typically the target to whose location information the rules apply.

M

mediation primitives
Program components that can be assembled into customized message-processing flows in conjunction with the IBM WebSphere Telecom Web Services Server (TWSS) Access Gateway.

message-driven bean (MDB)
An enterprise bean that provides asynchronous message support and clearly separates message and business processing.

mixed-media multilink transmission group (MMMLTG)
A multilink transmission group that contains links of different medium types (for example, token-ring, switched SDLC, nonswitched SDLC, and frame-relay links).

MLP Mobile Location Protocol, an Open Mobile Alliance (OMA) specification.

N

natural language support (NLS)
The ability for a user to communicate with hardware and software products in a language of choice to obtain results that are culturally acceptable.

O

Open Mobile Alliance (OMA)
A standards body that develops open standards for the mobile phone industry.

P

Parlay A set of specifications for application programming interfaces (APIs) for managing network services such as call control, messaging, and content-based charging.

Parlay Connector
A Parlay Connector is the primary system component of Telecom Web Services Server (TWSS) that provides connectivity to a Parlay gateway by using a distributed communication protocol, most commonly Common Object Request Broker Architecture (CORBA).

Parlay gateway
A server that hosts the service implementations for the Parlay API. The TWSS Parlay Connector communicates with the Parlay gateway over CORBA. The Parlay API consists of various telecom service APIs which provide an abstract interface to network elements deployed in the service provider network. Some TWSS Web service implementations utilize the Parlay Connector to enable using the Parlay API to support the functions exposed as Parlay X Web services.

Parlay X
A set of Web services designed to enable software developers to use telecommunication capabilities in applications.

Presence
A Parlay X Web service that allows client applications to use Web services to subscribe to a presentity, synchronously query the current presence information for a presentity, receive asynchronous notifications about changes in the presence information for a presentity, and unsubscribe from a presentity.

presence agent (PA)
A SIP user agent that is capable of receiving SUBSCRIBE requests, responding to them, and generating notifications of changes in presence state. A presence agent must have knowledge of the presence state of a presentity. This means that it must have access to presence data manipulated by PUAs for the presentity.

presence information
Information comprising one or more presence tuples.

presence server
A service that accepts, stores, and distributes presence information.

presence tuple
A set of data comprising a status, an optional communication address, and optional other presence information.

presence user agent (PUA)
A SIP user agent that manipulates presence information for a presentity. This manipulation can be the side effect of some other action (such as sending a SIP REGISTER request to add a new Contact) or can be done explicitly through the publication of presence documents. A presentity can have one or more PUAs. This means that a user can have many devices

(such as a cell phone and personal digital assistant (PDA), each of which is independently generating a component of the overall presence information for a presentity. PUAs push data into the presence system but are outside it; they do not receive SUBSCRIBE messages or send NOTIFY messages.

presentity
A presence entity, a software entity that provides presence information to a presence service.

public switched telephone network (PSTN)
A communications common carrier network that provides voice and data communications services over switched lines.

Q

R

registrar server
An SIP server that keeps track of where a user can be contacted and provides that information to callers. A SIP phone must register its current location with a registrar server to allow calls to be made to it using a phone number or alias. Without a registrar server, the caller would need to know the correct IP address and port of the telephone.

resource list server (RLS)
A server that accepts subscriptions to resource lists and sends notifications to update subscribers of the state of the resources in a resource list.

S

Service Component Architecture (SCA)
A set of specifications, published by the Open Service Oriented Architecture collaboration (OSOA), that describe a model for building applications and systems that builds on Service-Oriented Architecture (SOA) specifications.

Service Data Object (SDO)
An open standard for enabling applications to handle data from heterogeneous data sources in a uniform way. SDO incorporates J2EE patterns but simplifies the J2EE data programming model.

Service Policy Manager
A component of WebSphere Telecom Web Services Serverthat provides a storage capability and access mechanism to enable the definition of requesters, services, and subscriptions that associate services with requesters.

service-oriented architecture (SOA)
A conceptual description of the structure of a software system in terms of its components and the services they provide, without regard for the underlying implementation of these components, services and connections between components.

serving-call session control function (S-CSCF)
A server that acts as the central node of the signalling plane in a SIP network to register users and determine routing of messages. The S-CSCF also performs additional functions like providing routing services, enforcing policies, and providing billing information.

servlet
A Java program that runs on a Web server and extends the server's functionality by generating dynamic content in response to Web client requests. Servlets are commonly used to connect databases to the Web.

Session Initiation Protocol (SIP)
An Internet Engineering Task Force (IETF) standard protocol for initiating an interactive user session that involves multimedia elements such as video, voice, chat, gaming, and virtual reality.

Short Message Peer-to-Peer Protocol (SMPP)
A telecommunications industry protocol for exchanging Short Message Service (SMS) messages between SMS peer entities such as short message service centers.

Short Message Service (SMS)
A service that is used to transmit text to and from a mobile phone.

Simple Object Access Protocol (SOAP)
A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP can be used to query and return information and invoke services across the Internet.

SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE)
An architecture for the implementation of a traditional buddylist-based instant messaging and presence application with SIP.

stateless SIP proxy
A proxy that receives SIP requests and forwards the request to a particular SIP container in a cluster, based on SIP dialog affinity, load balancing, and failover considerations.

T

target (1) The destination for an action or operation. (2) An entry point into Partner Gateway. It is an instance of a receiver configured for a particular deployment; each target supports documents sent using a single transport type and multiple targets can exist for a given transport type, one for each document format. See also receiver.

Telecom Web Services Access Gateway
Provides policy-driven traffic monitoring, message capture, authorization, and management capabilities. These services are provided at the application layer, and they are enforced for each Web service request using knowledge of the requester, target service, and invoked operation.

WebSphere Telecom Web Services Server (TWSS)
WebSphere Telecom Web Services Server provides a middleware infrastructure for managing Web service access and an environment for hosting Web service API implementations, which provides flexibility for construction of tailored message processing logic in accordance with service provider network policies.

Terminal Location
A component of WebSphere Telecom Web Services Server that enables applications to send Web services requesting the Terminal Location services defined by the Parlay X 2.1 specification, and to register for Terminal Location Notifications.

Third Party Call
A Parlay X Web service that provides the ability to initiate a call from a network entity between two different users or user agents

time division multiple access (TDMA)
A technology for shared-medium (usually radio) networks. It allows several users to share the same frequency by dividing it into different time slots. The users transmit in rapid succession, one after the other, each using their own timeslot. This lets multiple users share the same transmission medium (for example, radio frequency) while using only the part of its bandwidth they require. In radio systems, TDMA is almost always used alongside frequency division multiple access (FDMA) and frequency division duplex (FDD); the combination is referred to as FDMA/TDMA/FDD.

U

Universal Mobile Telecommunications System (UMTS)
The third generation mobile telecommunications standard, defined by the ITU, that increases transmission speed to 2 Mbps per mobile user and establishes a global roaming standard.

user agent client (UAC)
In SIP, a client application that initiates the SIP request.

V

vertical cluster
A cluster in which the cluster members exist on a single physical server. Vertical clustering can be an effective way to take full advantage of a multiprocessor server. See also horizontal cluster.

W

W-CDMA (wideband code division multiple access)
A wideband spread-spectrum 3G mobile telecommunication air interface that uses CDMA. W-CDMA is the technology behind UMTS and is one of the interfaces used in cellular networks.

Web Services Description Language (WSDL)
An XML-based specification for describing networked services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

WebSphere Integration Developer (WID)
An integrated development and test environment and can be used as a visual editor when working with WebSphere Telecom Web Services Server mediation primitives to create customized flows.

WebSphere software for Telecom (WsT)
An IBM product suite that extends the industry leading WebSphere Application Server platform to deliver a fully IMS standards-compliant SIP application server, helping customers develop and deploy IP Multimedia Subsystem (IMS) compliant applications.

X

XCAP server
An HTTP server that acts as a repository for collections of XML documents. It manipulates user data such as authorization policy, resource list, and other XML resources and provides access to these resources through the HTTP protocol.

XML Configuration Access Protocol (XCAP)
An IETF specification (RFC 4825) that allows a client to read, write, and modify application configuration data stored in XML format on a server.

XML Document Management (XDM)
An OMA specification for accessing and manipulating XML documents that are stored in repositories in a network. Using XDM, an application can work with individual XML elements and attributes instead of entire documents. The XDM specification is based on the IETF XML Configuration Access Protocol (XCAP).

Y

Z

Numerics

3rd Generation Partnership Project (3GPP)
A collaboration agreement established in December 1998 through which ETSI (Europe), ARIB/TTC (Japan), CCSA (China), ATIS (North America), and TTA (South Korea) are making a globally applicable third-generation (3G) mobile phone system specification within the scope of the ITU's IMT-2000 project. 3GPP specifies the standards for UMTS.

3rd Generation Partnership Project 2 (3GPP2)
A collaboration agreement established in December 1998 through which ARIB/TTC (Japan), CCSA (China), TIA (North America), and TTA (South Korea) are making a globally applicable third-generation (3G) mobile phone system specification within the scope of the ITU's IMT-2000 project. 3GPP2 specifies the standards for CDMA2000.

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 grant 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.

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 may 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.

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.

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.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
DB2
IBM
pSeries
Tivoli
WebSphere

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.

Other company, product, or service names may be trademarks or service marks of others.

Readers' Comments — We'd Like to Hear from You

IBM WebSphere Presence Server
IBM WebSphere Presence Server
Version 7.0

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: 1-800-227-5088 (US and Canada)

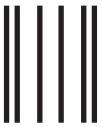
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department 6R4A
P.O. Box 12195
Research Triangle Park, NC 27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line