WebSphere® IBM WebSphere XML Document Management Server

Version 7.0

IBM WebSphere XML Document Management Server

SC00-0000-00

WebSphere® IBM WebSphere XML Document Management Server

IBM

Version 7.0

IBM WebSphere XML Document Management Server

SC00-0000-00

Note
Before using this information and the product that it supports, read the information in "Notices" on page 149.

Second Edition (April 2010)

This edition applies to IBM WebSphere XML Document Management 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

Welcome to the IBM® WebSphere® XML Document Management Server Component (IBM XDMS) information center. The following links will help you access the available information.

Introducing the IBM WebSphere XML Document Management Server Component

The IBM WebSphere XML Document Management Server Component (XDMS) provides storage, management and subscription to documents that are owned by entities within your IMS-based solution.

IBM XDMS is based on the Open Mobile Alliance (OMA) standards for the *OMA XDM v2.0* specification. The OMA XDMS specification defines the use of two IETF-based network specifications for interaction with the XDMS:

XML Configuration Access Protocol (XCAP)
: XCAP protocol is used to retrieve, delete, and manipulate XML documents over HTTP.

Session Initiation Protocol (SIP)
: SIP allows clients to subscribe and receive notifications when a document is added, deleted, or updated.

The following figure illustrates the way in which data flows through the IBM XDMS product.

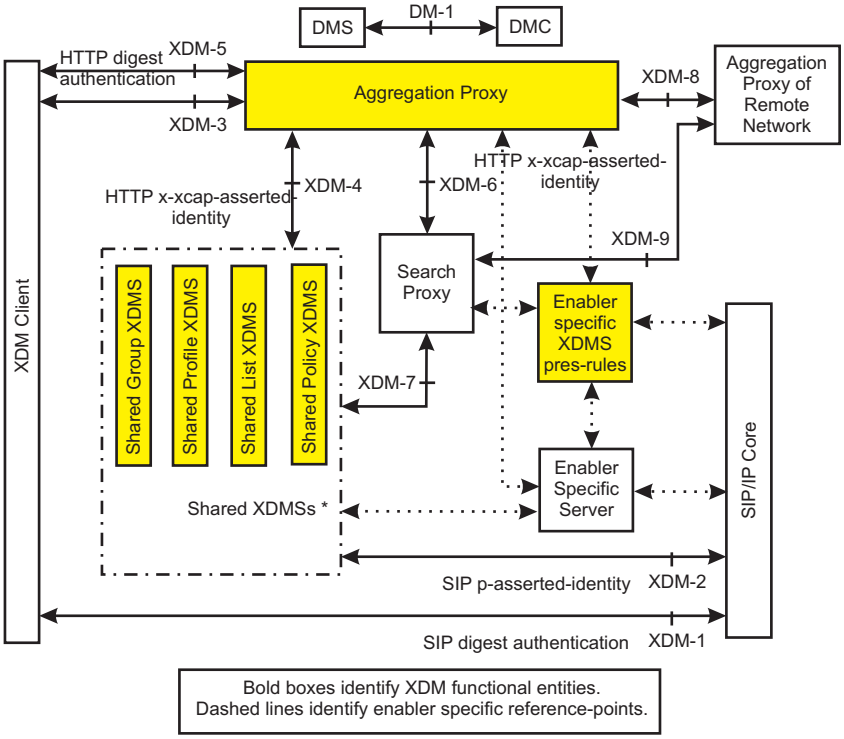
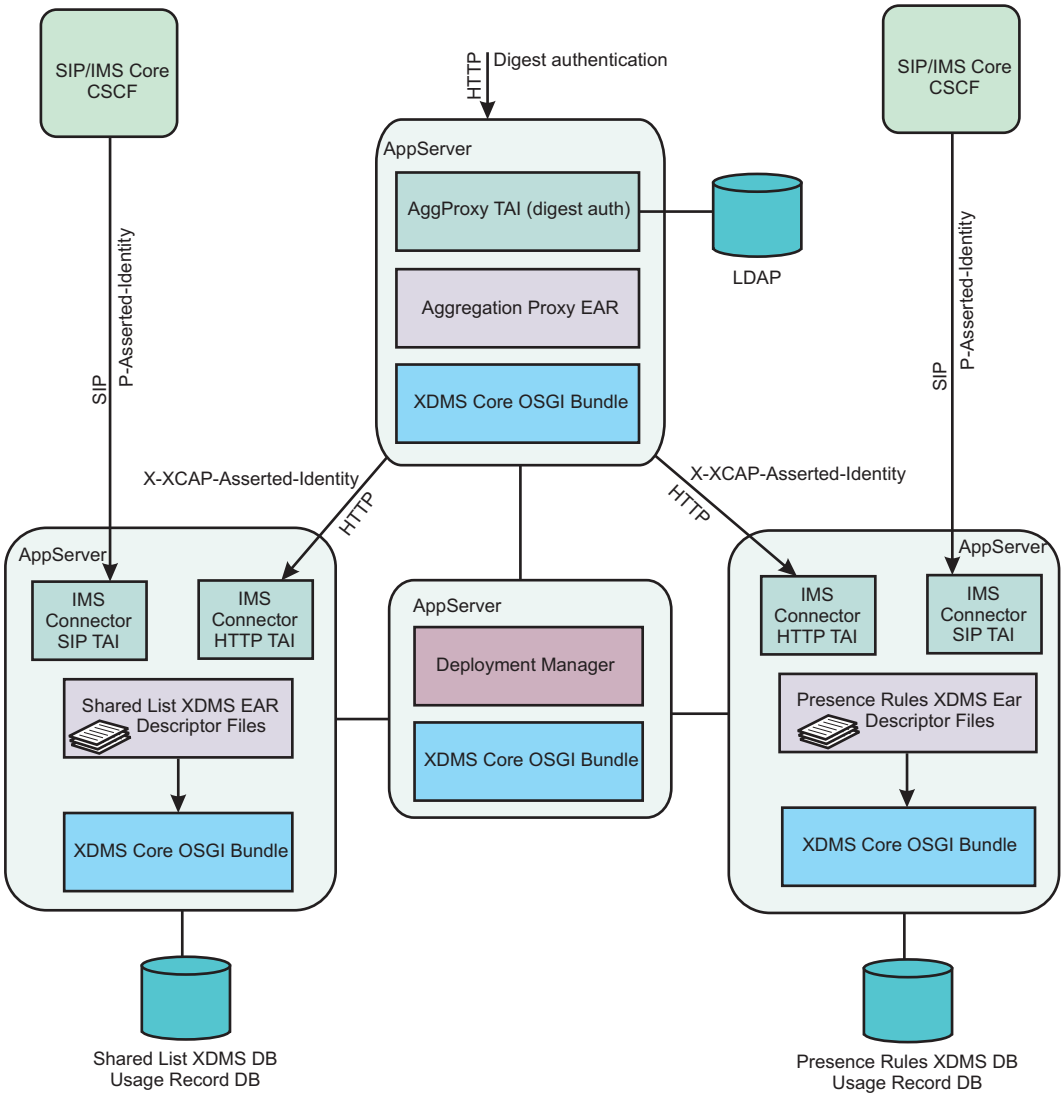


Figure 1. Data flow through the IBM XDMS product

The following figure illustrates the way in which IBM XDMS components interact with each other. In this example, separate enablers are deployed for SharedList XDMS and Presence Rules. Additional enablers can be added as well.



OMA 2.0 enablers are provided that enable applications to store XML content based on the application usages as described for IBM XDMS:

- A SharedList XDMS for managing resource-lists and rls-services documents that are common to other IMS components.
- A PresenceRules XDMS for handling presence rules associated with protecting presence information.
- A SharedProfile XDMS for storing user profiles.
- A SharedPolicy XDMS for storing push to talk (POC) authorization policies.
- A SharedGroup XDMS for storing POC group service definitions.

An Aggregation Proxy is provided to route requests based on the application usage to the corresponding IBM XDMS enablers. It can also aggregate capabilities and directory information from multiple IBM XDMS enablers. The Aggregation Proxy is considered the client facing service that is a single point of entry to access all IBM XDMS enablers.

The XDM client authenticates by means of digest authentication against the Aggregation Proxy (HTTP) and the IMS Core (SIP). (In digest authentication, encryption is used so that a user's credentials can be established without the need

to transmit a password in plaintext over the network.) Asserted identities are then used to forward requests with the identities within the trusted environment. The Aggregation Proxy can be configured to forward requests to an external aggregation proxy. The external aggregation proxy is simply treated as another enabler

The IBM XDMS product can be customized through the development of custom XDMS enablers allowing third parties to develop custom code that provide constraints for specific needs. Additionally, system administrators are given highly-granular control of enablers through configurable properties. This flexibility allows the IBM XDMS to be tailored for the specific needs required in various telecommunication environments.

Digest authentication using a WebSphere Trust Association Interceptor (TAI) and standard support for Transport Layer Security (TLS) are provided to enhance security when accessing the Aggregation Proxy.

IBM XDMS features

The IBM WebSphere XML Document Management Server Component (XDMS) is highly extensible and has many useful features.

Aggregation Proxy

The Aggregation Proxy provides a single point of entry to access multiple IBM WebSphere XML Document Management Server Component (XDMS) enablers.

The Aggregation Proxy provides a single point of entry to access multiple IBM XDMS enablers that are configured in the backend. Therefore, it can be considered to have the superset of capabilities for all IBM XDMS enablers. This simplifies the client configuration in that all XCAP communication is sent to a single entity, which is the Aggregation Proxy.

Features

IBM Aggregation Proxy fully supports Open Mobile Alliance (OMA) specification, and offers additional functionality as well.

Additional functionality includes:

- Domain routing
- Enhanced multithreaded directory performance
- Added security
- Configurable xcap-caps optimizer
- Configurable fault alarm mechanism

Uses of the Aggregation Proxy

Authentication and authorization

A single XDMS environment can be used to host users which are logically separated into different domains. It can be important to physically separate the storage and manipulation of users' documents for a number of reasons including:

- XDMS performance
- machine utilization
- security

- database limitations

To respect these needs, the Aggregation Proxy allows users to be separated by domain, routing requests accordingly based on administrator-defined rules. This allows back-end XDMS utilization to be highly configurable at a single point of administration.

Enhanced multi-threaded directory performance
Without the Aggregation Proxy, clients desiring to access the directory of a user's documents across all available XDMSs would need to request each directory document serially then aggregate the results.

The Aggregation Proxy simplifies this task. With the Aggregation Proxy in place, to obtain a directory document containing all of a user's documents, a client only needs to query the Aggregation Proxy for the user's `directory.xml` document.

The process of querying each XDMS and aggregating the results into a single document is handled by the Aggregation Proxy. The Aggregation Proxy contacts each backend XDMS in parallel. This improves performance drastically over client-side implementations reducing latency for each directory request.

Security
Aggregation Proxy can partition users according to security policies and domain routing configuration.

Aggregation Proxy Trust Association Interceptor

The Aggregation Proxy Trust Association Interceptor is an HTTP Digest TAI: It provides digest authorization, which takes precedence over basic authorization. (In digest authentication, encryption is used so that a user's credentials can be established without the need to transmit a password in plaintext over the network.)

The HTTP digest TAI uses Lightweight Directory Access Protocol (LDAP) for its authorization implementation and passes a p-asserted-identity header to the back-end XDMS for authentication.

Security

The IBM WebSphere XML Document Management Server Component (XDMS) has a number of security features.

The following section lists each security component:

IMS Trust Association Interceptor
The Trust Association Interceptor (TAI) is common to all IMS components and enables trust association for both Session Initiation Protocol (SIP) and HTTP.

Aggregation Proxy Trust Association Interceptor
IBM has developed a separate trust association interceptor designed specifically to work with Aggregation Proxy.

WebSphere Application Server Security
An administrator user account must be added on your WebSphere Application Servers to secure them.

Digest Authentication
Aggregation Proxy uses digest authentication to authenticate users. (In

digest authentication, encryption is used so that a user's credentials can be established without the need to transmit a password in plaintext over the network.) You need to set up a user repository and configure Aggregation Proxy to use it. WebSphere allows for the use of the following user repositories:

- Lightweight Directory Access Protocol (LDAP)
- custom user registries
- file-based

XDMS enablers

IBM XDMS provides support for several XDMS enablers that are based on specifications set forth by the Open Mobile Alliance (OMA).

Shared List XDMS

The Shared List XDMS enabler describes two types of shared lists: the URI List and the Group Usage List (a list of group names or service URIs that are known by the XDMS).

The Shared List XDMS is a server entity that supports the following functions:

- Manages and supports the content of URI List and Group Usage List XML documents
- Performs authorization of incoming Session Initiation Protocol (SIP) and XML Configuration Access Protocol (XCAP) requests
- Notifies subscribers of changes in XML documents
- Provides aggregation of notifications of changes to multiple XML documents

Shared List XDMS contains shared list applications, referred to as Application Unique Identity Descriptors (AUIDs). The following AUIDs are defined:

resource-lists
Allows end users to create shared documents used to store lists of members. Standard support for XCAP and SIP are provided for resource-lists documents.

rls-services
Allows end users to create RLS documents that can either store embedded lists or reference a local resource-lists document that is stored within the Shared List XDMS. Standard support for XCAP and SIP are provided for rls-services documents.

org.openmobilealliance.group-usage-list
The *OMA-TS-XDM_Shared_List-V2_0* specification provides the ability to store group usage list documents. This AUID was added to the Shared List XDMS enabler to support group usage lists.

com.ibm.resource-lists-acls
The access control list AUID for `resource-lists`.

com.ibm.rls-services-acls
The access control list AUID for `rls-services`.

com.ibm.group-usage-list-acls
The access control list AUID for `org.openmobilealliance.group-usage-list`.

Presence Rules XDMS

Presence Rules XDMS supports the Presence Rules document storage and XDMS document subscribe and notify functions.

The following Application Unique Identity Descriptors (AUIDs) are defined for Presence Rules:

org.openmobilealliance.pres-rules
Allows Session Initiation Protocol (SIP) subscriptions to change notification by referencing the document XML Configuration Access Protocol (XCAP) URI in the document event header.

com.ibm.group-usage-list-acls
The access control list AUID for `org.openmobilealliance.pres-rules`.

Shared Profile XDMS

The Shared Profile XDMS is the logical repository for user profile documents.

The Shared Profile XDMS is a server entity that supports the following functions:

- Manages and supports the content of User Profile XML documents
- Performs authorization of incoming Session Initiation Protocol (SIP) and XML Configuration Access Protocol (XCAP) requests
- Notifies subscribers of changes in XML documents
- Provides aggregation of notifications of changes to multiple XML documents
- Provides search results

Shared Profile XDMS contains Application Unique Identity Descriptors (AUIDs) for both a public user profile and a private locked user profile. Every user profile maps a person's XUI through the URI attribute to various types of information such as first or last name, email address, location, hobbies, or anything else that someone would like to share.

The following AUIDs are defined for Shared Profile:

org.openmobilealliance.user-profile
Used to store public user profiles.

org.openmobilealliance.locked-user-profile
Used to store private (locked) user profiles.

com.ibm.user-profile-acls
The access control list AUID for `org.openmobilealliance.user-profile`.

com.ibm.locked-user-profile-acls
The access control list AUID for `org.openmobilealliance.locked-user-profile`.

Shared Policy XDMS

The Shared Policy XDMS is the logical repository for storing push to talk (POC) authorization policies.

The Shared Policy XDMS is a server entity that supports the following functions:

- Manages and supports the content of user access policy XML documents
- Performs authorization of incoming Session Initiation Protocol (SIP) and XML Configuration Access Protocol (XCAP) requests
- Notifies subscribers of changes in XML documents
- Provides aggregation of notifications of changes to multiple XML documents

The following Application Unique Identity Descriptors (AUIDs) are defined for Shared Policy:

org.openmobilealliance.access-rules
Used to store POC authorization policies.

com.ibm.access-rules-acls
The access control list AUID for `org.openmobilealliance.access-rules`.

Note: POC version 1.0 clients (PoC_XDM-V1_0) can also use the Shared Policy XDMS to store the old version of the POC authorization policies. However, IBM XDMS version 7.0 does not support such capability. Only the new Shared Policy AUID, `org.openmobilealliance.access-rules`, is supported.

Shared Group XDMS

The Shared Group XDMS is the logical repository for Group documents. It reuses the push to talk (POC) Group document structure to make the syntax backward compatible with that of the POC Group document.

The Shared Group XDMS is a server entity that supports the following functions:

- Manages and supports the content of Group XML documents
- Performs authorization of incoming Session Initiation Protocol (SIP) and XML Configuration Access Protocol (XCAP) requests
- Notifies subscribers of changes in XML documents
- Provides aggregation of notifications of changes to multiple XML documents
- Provides search results

The following Application Unique Identity Descriptors (AUIDs) are defined for Shared Group:

org.openmobilealliance.groups
Used to store POC Group service definitions.

com.ibm.groups-acls
The access control list AUID for `org.openmobilealliance.groups`.

Note: POC version 1.0 clients (PoC_XDM-V1_0) can use the Shared Group XDMS to store the old version of the POC group. However, IBM XDMS version 7.0 does not support such capability. Only the new Shared Group AUID, `org.openmobilealliance.groups`, is supported.

Shared Enablers XDMS

The Shared Enablers XDMS is a single EAR file that contains all of the Application Unique Identity Descriptors (AUIDs) from all of the other XDMS enablers.

The Shared Enablers XDMS has all of the same functionality as the other enablers, in one EAR file. The benefit of this enabler is that you do not have to set up multiple clusters or data sources for each enabler. Also, the Aggregation Proxy is not needed when using this enabler by itself, because aggregation- or enabler-specific routing is not necessary. The Aggregation Proxy is still needed, however, if domain routing is desired.

Supported SIP subscription event types

IBM XDMS supports two types of Session Initiation Protocol (SIP) subscriptions.

The IBM XDMS supports subscriptions using UA-profile or xcap-diff events. While both are supported, xcap-diff events have several advantages over UA-profile events. Both UA-profile and xcap-diff events support change notification, with some differences.

UA-profile events

The IBM XDMS supports **UA-profile** events as defined in the *OMA-TS-XDM_Core-V1_0_1* specification, which can be used to subscribe to XDMS documents/directories.

Xcap-diff events

IBM XDMS 7.0 also supports xcap-diff events as defined in the *OMA-TS-XDM_Core-V2_0* specifications. xcap-diff events can be used for the following tasks:

- Subscribe to documents and directories
- Subscribe to document elements and attributes
- Send batch subscriptions
- Subscribe to future resources before they are created

Mapping and normalization

The mapping and normalization feature is used when mapping multiple public user identities to one private identity.

Multiple public ID mapping

With the vast array of communication devices and contexts for which we use these devices, it is not uncommon to have more than one user identity for the various devices. Although a person can have multiple public user identities, the person behind the public user identities is still the same and there are situations where it is useful to map the public identities to one primary private identity.

There are many scenarios where the mapping should take place, for example access control. If you are granted access to a document with one public identity, you should be allowed to access that document with another public identity. For customers wishing to implement multiple public identity mapping, the IBM WebSphere XML Document Management Server Component (XDMS) provides the following:

- An interface method to compare two public/private identities to determine equivalence.
- An interface method to return the private identity from a public identity.

Implementation of multiple public identity mapping will be customer dependant. Samples and more information on developing implementations are provided in the Telecom Application Enablement Feature and Telecom Application Enablement Feature Information Center.

Normalization

Unique identifiers (UID) are normalized for consistency within the IBM XDMS. The way IBM XDMS performs normalization is defined by a set of properties.

When a UID is used to specify a document or directory, IBM XDMS normalizes the UID. This includes UIDs embedded in document selectors, as well as, whenever a UID is used to specify a user directory or match a service URI.

XDMS extensibility

Add support for a new application usage or update the XML Configuration Access Protocol (XCAP) processing for an existing application usage.

The IBM XDMS allows customers to easily add support for a new application usage or update the XCAP processing for an existing application usage. The two primary interfaces provided by an XDMS are XCAP and SIP, both of which are able to be modified by adding new application support or customization of current application support.

More information on customizing IBM XDMS applications or developing your own custom applications can be found in the Telecom Application Enablement Feature Information Center.

Chapter 2. Planning

The following topics describe the deployment choices that are available to you as you plan for the IBM WebSphere XML Document Management Server Component (IBM XDMS).

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 12

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. IBM XDMS also requires an LDAP server if you intend to deploy it with the Aggregation Proxy.

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"
"Java version" on page 13
"Databases" on page 13
"LDAP server" on page 13

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 11.1.0.7

LDAP server

XDMS enablers and the IBM XDMS Aggregation Proxy retrieve users from an LDAP server. You can use any LDAP v3-compliant server for this purpose.

The IBM Tivoli® Directory Server (ITDS) is an example of such an LDAP server. Your WebSphere Application Server product license grants permission to install and use one copy of ITDS.

System requirements

The end-to-end system consists of two cooperating components, both of which are specified by the OMA v2.0 specification. The components are the Aggregation Proxy and the IBM WebSphere XML Document Management Server Component (IBM XDMS).

Both component subsystems run in a WebSphere Application Server Network Deployment environment and can be configured on separate physical servers and clusters to provide scaling flexibility. IBM XDMS requires a DB2 or Oracle database for storage of user documents. IBM XDMS also requires a JMS-based messaging engine for communication within the cluster.

Note: You can deploy a standalone XDMS subsystem if you do not require the functionality of the Aggregation Proxy.

Evaluating your hardware environment

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

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 IBM XDMS, only clustered environments are supported.

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. HTTP redirector products can also be used to implement horizontal scaling. 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 IMS™ Application Server applications. This toolkit is available as a free download. It is designed to reduce the time to develop applications that use IBM XDMS and other IBM WebSphere software for Telecom program products.

Planning your IBM XDMS deployment

The IBM WebSphere XML Document Management Server Component (XDMS) can be deployed in a clustered environment or on a single application server.

Clustered deployment of IBM XDMS

Installation of IBM XDMS in a clustered WebSphere Application Server environment is the recommended deployment configuration.

Clustered deployments have additional requirements and considerations compared to the standalone deployment. The following requirements apply for clustered deployments:

- Clusters must be created for each IBM XDMS application that you plan to install.

- There must be a data store for the JMS SIBus application.

You should also consider the various hardware topologies that are available, paying special attention to the scalability considerations that are described in the topic *Evaluating your hardware environment*.

Cluster management

As clusters, both the IBM XDMS and the Aggregation Proxy must be fronted with some network element that performs routing to these clusters and handles request processing. For Session Initiation Protocol (SIP) traffic, the XDMS cluster must have a highly available set of WebSphere Application Server proxies to do the request routing. This set of proxies is itself clustered and is fronted by some type of IP sprayer or load balancer.

For HTTP traffic destined for the Aggregation Proxy or the IBM XDMS, either the WebSphere Application Server Proxy or some equivalent HTTP router such as IBM HTTP Server or an IP sprayer can be used. In addition, the proxy/router fronting the Aggregation Proxy cluster must be configured to ensure affinity based on client IP or similar if retry counts on failed authentication attempts are used.

The following diagrams depict typical highly available configurations for Aggregation Proxy and IBM XDMS:

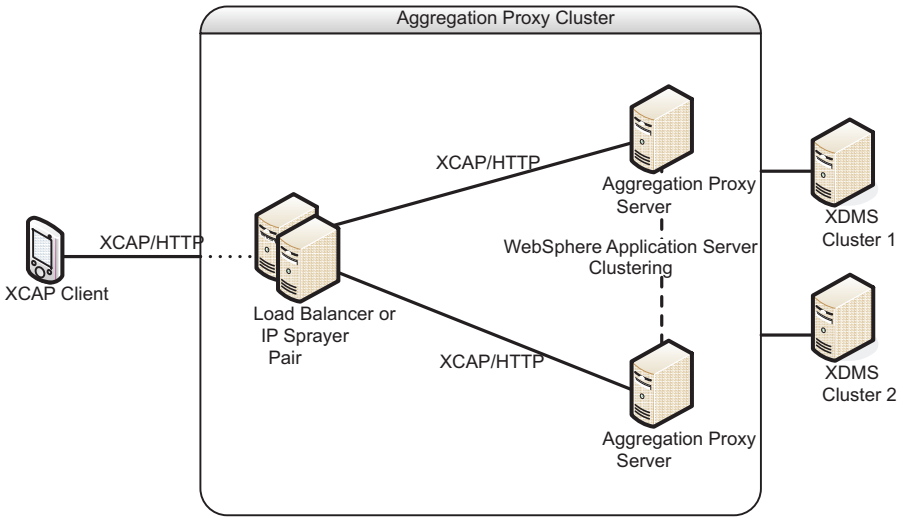


Figure 2. High availability Aggregation Proxy

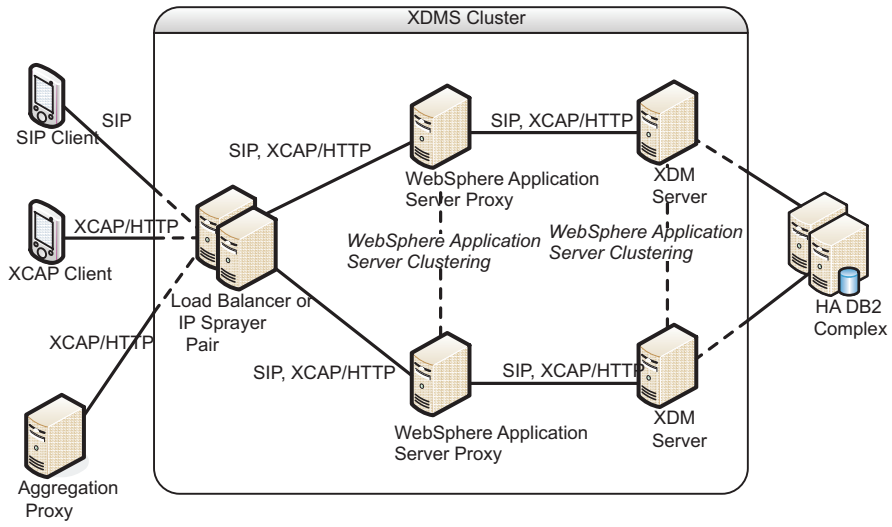


Figure 3. High availability XDMS

High availability and scaling

The IBM XDMS components use WebSphere Application Server clustering technology to achieve a highly available and scalable architecture. The Aggregation Proxy and the IBM XDMS are both independently scalable because they are both deployed in separate clusters.

In addition to WebSphere Application Server, both the Aggregation Proxy and the IBM XDMS provide core functionality that can enhance the scalability of the end-to-end solution. You can deploy clusters for the IBM XDMS in different combinations of domain and Application Unique ID (AUID).

Application Usage Identifier partitioning

The AUID supported by the system can be hosted all on the same IBM XDMS or subsets of them can be hosted on a dedicated IBM XDMS or IBM XDMS cluster. By partitioning to separate clusters, the overall scalability and supportable throughput as a whole is improved. For example, you can use a separate high-availability cluster and database for your resource-lists documents from your presence rules documents. Aggregation Proxy request forwarding and routing is used to control to which IBM XDMS requests are sent.

Domain partitioning

Using Aggregation Proxy request forwarding, you can configure different IBM XDMS instances to support or be dedicated to certain subscriber domains. This capability enables supporting multiple user space partitions when the total user population is too large for a single IBM XDMS cluster, or when there are different security requirements for different domains.

AUID partitioning can be used in conjunction with domain partitioning whereby the Aggregation Proxy first determines the subset of XDMS applications supporting the AUID and then discriminates between that subset for the domain matching the XUI of the requested user.

The following is a diagram of a typical deployment that uses both partitioning techniques for scaling:

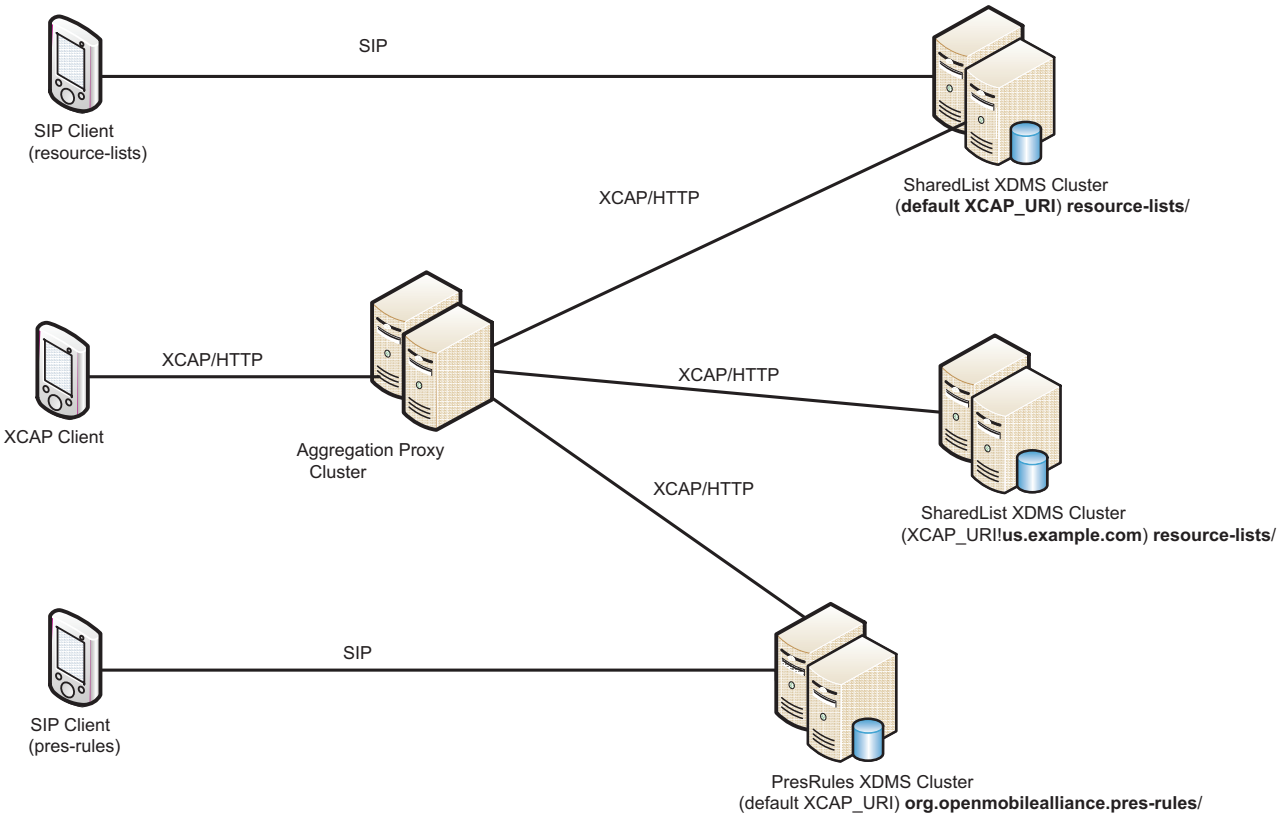


Figure 4. Subscriber and AUID partitioning

In this example, pres-rules and resource-lists are partitioned across three different clusters, where resource-lists are additionally partitioned across two domains (a default domain and a specific domain called "us").

Note: SIP clients, which can be the same physical device as the XML Configuration Access Protocol (XCAP) client are aware of the specific IBM XDMS cluster endpoint address, whereas the XCAP client is aware of the Aggregation Proxy endpoint only.

Remember that the clustering techniques for high-availability and scaling can also be combined with the partitioning techniques.

Installation prerequisites for clustered deployments

Before moving on to installation make sure you have met the following prerequisites:

- WebSphere Application Server version 6.1.0.x or 7.0.0.1 is installed with a deployment manager with at least one managed node.
- You have federated all nodes into the deployment manager cell.
- A WebSphere user account repository is created. A federated Lightweight Directory Access Protocol (LDAP) repository is recommended. Refer to the WebSphere Application Server 7.0 Information Center for more instructions.

- DB2 The IBM DB2 Enterprise Server Edition version 9.5 FixPak 1 server has been installed along with the license for pureXML®.
- The IBM DB2 Enterprise Server Edition version 9.5 FixPak 1 client JDBC library JARs have been installed in the same path on every WebSphere Application Server node in the cluster. This allows the WebSphere variables for the JDBC driver to be declared at the cell level.
- Oracle Oracle Database version 11.1.0.7 is installed.
- Oracle One of the following Oracle clients is installed on every WebSphere Application Server node in the cluster:

  Oracle Database Client version 11.1.0.7

  Oracle Database Instant Client Package - Basic, version 11.1.0.7

- You have created the clusters for each IBM XDMS application that you plan to install.

Standalone deployment of IBM XDMS

The standalone installation for IBM XDMS is recommended for development and testing purposes.

Standalone installation of IBM XDMS differs from clustered installations. The following describes some of the differences, standalone installation:

- Requires only an WebSphere Application Server application server profile.
- Does not require the prerequisite creation of clusters for theIBM XDMS applications.
- Can use file store instead of data store for JMS SIBus configuration.
- Aggregation Proxy is not necessary to install.

You should also consider hardware topologies, especially those described under the *Development topology* in the *Hardware topology* section of this information center.

Before moving on to installation make sure you have met the following prerequisites:

- WebSphere Application Server version 6.1.0.x or 7.0.0.1 is installed.
- A WebSphere user account repository is created. Refer to the WebSphere Application Server 7.0 Information Center for more instructions.
- DB2 The IBM DB2 Enterprise Server Edition version 9.5 FixPak 1 server has been installed along with the license for pureXML.
- DB2 The IBM DB2 Enterprise Server Edition version 9.5 FixPak 1 client JDBC library JARs have been installed on the WebSphere Application Server.
- Oracle Oracle Database version 11.1.0.7 is installed.
- Oracle One of the following is installed:

  Oracle Database Client version 11.1.0.7

  Oracle Database Instant Client Package - Basic, version 11.1.0.7

Planning for the IBM XDMS application

There are planning considerations for each IBM XDMS application that you install.

Planning for database capacity

You need to plan the amount of database space to dedicate to user document storage.

Several factors enter into this calculation:

- The number of expected subscribers, both at first and as your system grows over time: *Ns*.
- The average document size for each type of stored document: resource list (*Zr*), authorization (*Za*), presence rules (*Zp*), and global documents (*Zg*).
- The average number of documents per subscriber, per document type: *Dr* and *Dp*. There is typically only a single authorization document per subscriber.
- The number of physical global documents: *Ng*.

As an example, these factors can be combined in a simple formula to provide a rough assessment of the storage requirement: *Ns*((Dr*Zr)+Za+(Dp*Zp))+Ng*Zg*

You also need to plan the database-serving capacity as a function of Input/Output instructions per second (IOPS). To size your I/O serving capacity requirement, you need to determine the workload capacity for each disk in the array that supports the IBM XDMS database.

Assume that each 15K RPM disk in the array can sustain 120 IOPS. Use this rate combined with the expected workload on the database to derive the number and types of disks you will need to support the database. Information about the relationship of number of XML Configuration Access Protocol (XCAP) requests to I/O requests for typical workloads is documented in the *Capacity Planning Guide*, which you can obtain from the IBM Support Web site.

You can also configure IBM XDMS to generate usage records (audit/billing records) for all activity that it processes. You should plan for enough database capacity to handle the expected billing traffic within the limits and constraints imposed by your usage record processing policy (archival, extraction, and so forth). More details and recommendations are available in the *Capacity Planning Guide*.

Subscription support considerations

IBM XDMS supports both UA-profile and xcap-diff subscription events.

IBM XDMS supports subscription to service element, document and document elements, including bulk subscriptions. Some of the supported subscription levels require the implementation of xcap-dff events. The following table lists the subscription levels and required event types.

Table 1. Subscription levels

subscription level	**event type required**
Service element, in example, rls-services	xcap-diff
directory or document	UA-profile, xcap-diff
document element or attribute	xcap-diff

Subscribing to document elements requires that you implement xcap-diff event headers and define node level constraints in the Application Unique Identity Descriptors (AUID) descriptor file for each XDMS enabler. To modify enablers, refer to the *Telecom Application Enablement Feature* information center.

Planning for Aggregation Proxy routing

The Aggregation Proxy can be configured with basic or advanced routing options.

Basic Aggregation Proxy routing

Basic routing uses the list of IBM XDMS enablers configured in the resource environment providers to route requests to specific Application Unique Identity Descriptors (AUIDs).

The Aggregation Proxy provides a single point of entry to access multiple IBM XDMS enablers that are configured in the backend. Therefore, it can be considered to have the superset of capabilities for all IBM XDMS enablers. This simplifies the client configuration in that all XML Configuration Access Protocol (XCAP) communication is sent to a single entity, which is the Aggregation Proxy.

During the installation and configuration of the Aggregation Proxy, the list of IBM XDMS enablers is configured in the Resource Environment Provider (REP) property called XDMS_URI. The list must contain a unique set of IBM XDMS enablers in that no two XDMS enablers can support the same AUID (except for two common AUIDs, xcap-caps and org.openmobilealliance.xcap-directory, which all IBM XDMS enablers must implement). This is required so that for each AUID, there is a unique route to a specific IBM XDMS enabler. If two IBM XDMS enablers are defined to support the same AUID, then the Aggregation Proxy cannot determine which IBM XDMS enabler to forward requests to. The AUID, therefore, is rendered unserviceable.

Typically, the IBM XDMS enablers in the backend are secured by a firewall and should be considered trusted. However, if higher security is required and HTTPS is enabled for the IBM XDMS enablers, then you can set the REP property HTTPS_PROXY_XDMS to true, which indicates the use of encryption. Then, instead of defining your list of IBM XDMS enablers in the REP property XDMS_URI, define the list of XDMS enablers in the REP property XDMS_URI_HTTPS.

Optionally, the advanced routing topic discusses usage of domain partitioning to support multiple XDMS enablers with the same AUID but separated by domain. This allows the segregation of certain client populations to be routed to a specific IBM XDMS enabler for a specific domain. See the topic *Advanced Aggregation Proxy routing* for more information.

Basic routing example

The following Aggregation Proxy REP configuration is used for the examples in this topic:

1. `PROXY_ROOT = http://example.com/services`
2. `HTTPS_PROXY_XDMS= false`
3. `XDMS_URI = http://sharedlist1.com:9080/services#http://presrules1.com:9081/services`

The SharedList IBM XDMS enabler *sharedlist1.com* supports the AUID resource-lists. The PresenceRules IBM XDMS enabler *presrules1.com* supports the AUID org.openmobilealliance.pres-rules.

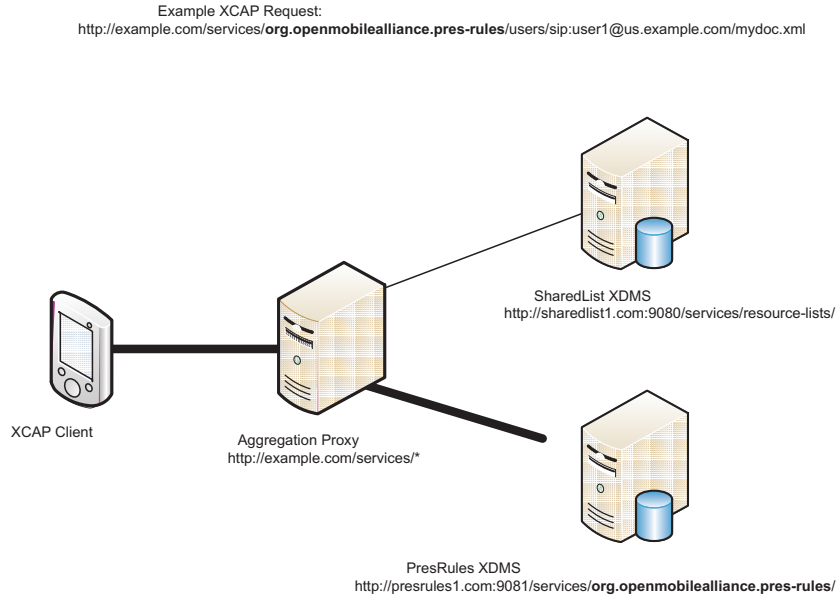


Figure 5. Basic routing diagram

Advanced Aggregation Proxy routing

Advanced routing uses domain partitioning and matching to determine the routing of XCAP requests.

Basic routing is simply based on the Application Unique Identity Descriptor (AUID) in order to determine the unique IBM XDMS enabler that the request is routed to. However, there may be situations where the user population is too large for a single IBM XDMS enabler to support or where there are security requirements to partition the users. Therefore, besides defining the default IBM XDMS enablers, an administrator may choose to define additional IBM XDMS enablers that support the same AUID but for different domains.

During the installation and configuration of the Aggregation Proxy, the list of IBM XDMS enablers for a specific domain is configured in the Resource Environment Provider (REP) property `XDMS_URI!<domain>` where the `<domain>` portion of the name defines the domain for which the list of IBM XDMS enablers is supported. The implication here is that two IBM XDMS enablers supporting the same AUID can be defined in the Aggregation Proxy if they reside in different domains. If for each AUID and domain combination, the request matches only one IBM XDMS enabler, then there is no ambiguity in routing.

The portion of the request that is used for domain matching is found in the XML Configuration Access Protocol (XCAP) User Identifier (XUI). The XUI is typically a URI such as a Session Initiation Protocol (SIP) URI. The URI may contain a domain which is parsed and then used to match on the appropriately configured XDMS enabler to route the request to. Matching is done first by the most qualified domain towards least qualified. If no domain is matched, then the default XDMS enabler is used as specified during basic routing configuration. If the XCAP request

is for a global document (no XUI), then no domain is specified and consequently the request is routed to the default XDMS.

Advanced routing examples

These examples illustrate the various routes given the domain and AUID matching done by the Aggregation Proxy.

The following Aggregation Proxy REP configuration is used for the examples in this topic:

1. `PROXY_ROOT = http://example.com/services`
2. `HTTPS_PROXY_IBM XDMS = false`
3. `XDMS_URI = http://sharedlist1.com:9080/services#http://presrules1.com:9081/services`
4. `XDMS_URI!us.example.com = http://sharedlist2.com:9080/services/resource-lists`

The SharedList IBM XDMS enablers sharedlist1.com and sharedlist2.com both support the AUID resource-lists. The PresenceRules IBM XDMS enabler presrules1.com supports the AUID org.openmobilealliance.pres-rules.

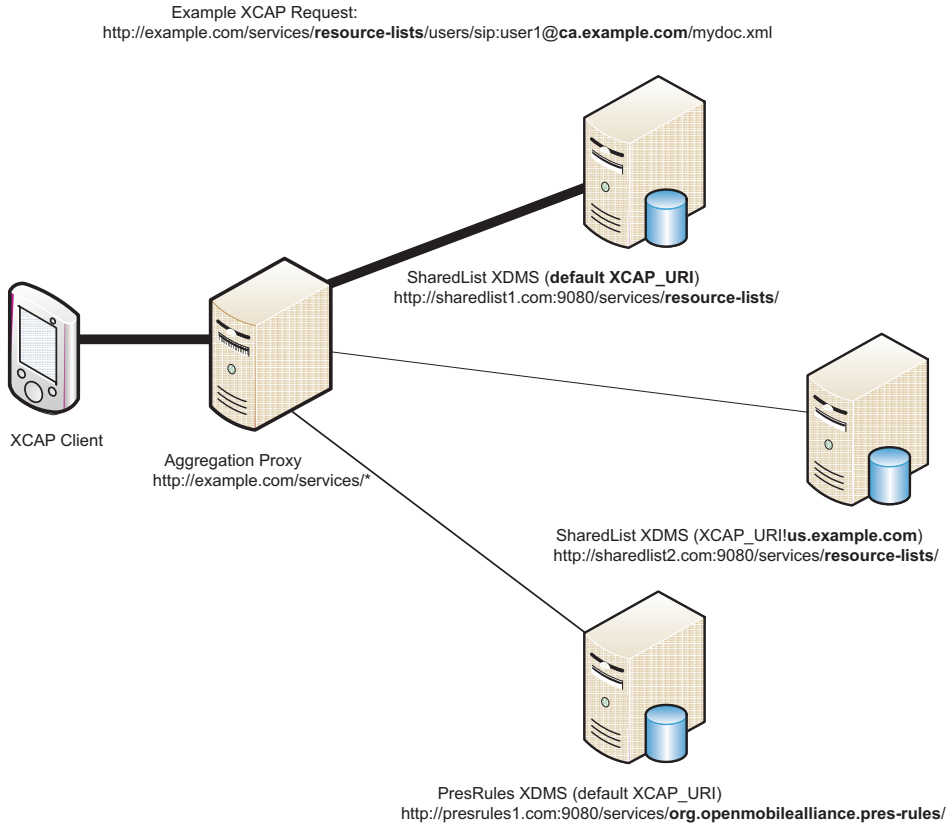


Figure 6. Advanced routing example one

The Aggregation Proxy routes to the default domain because it did not find a closer match to the XCAP Request XUI.

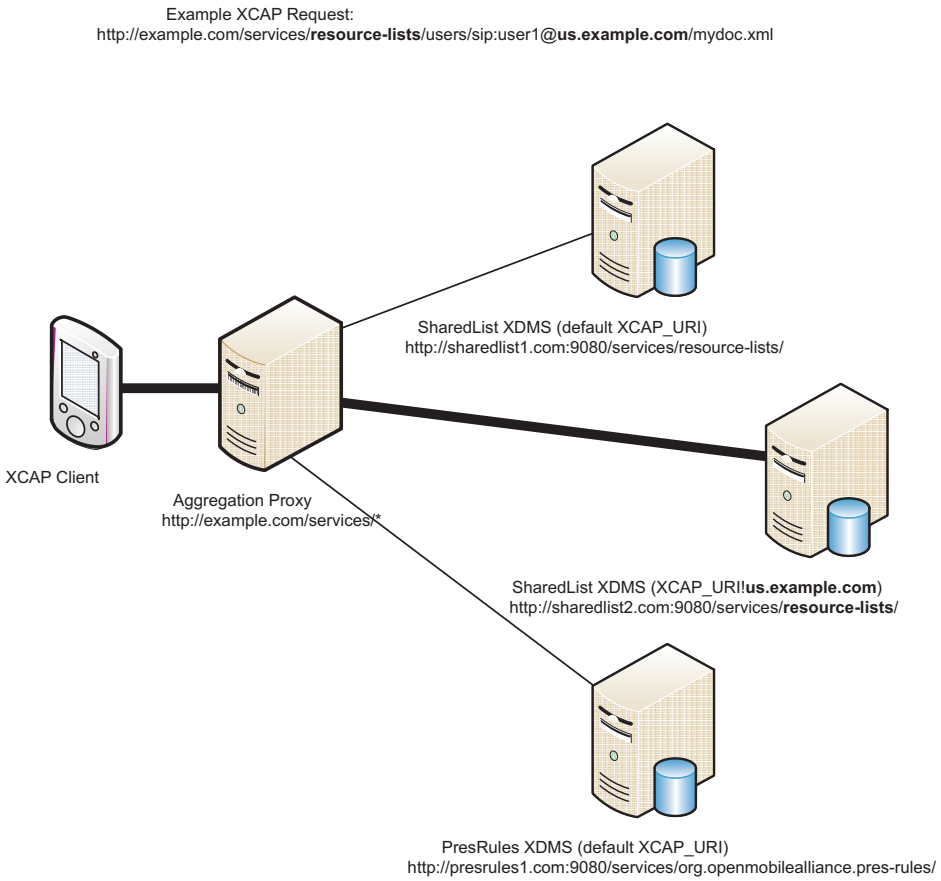


Figure 7. Advanced routing example two

The Aggregation Proxy routes to the domain `us.example.com` because it matches the XUI in the request.

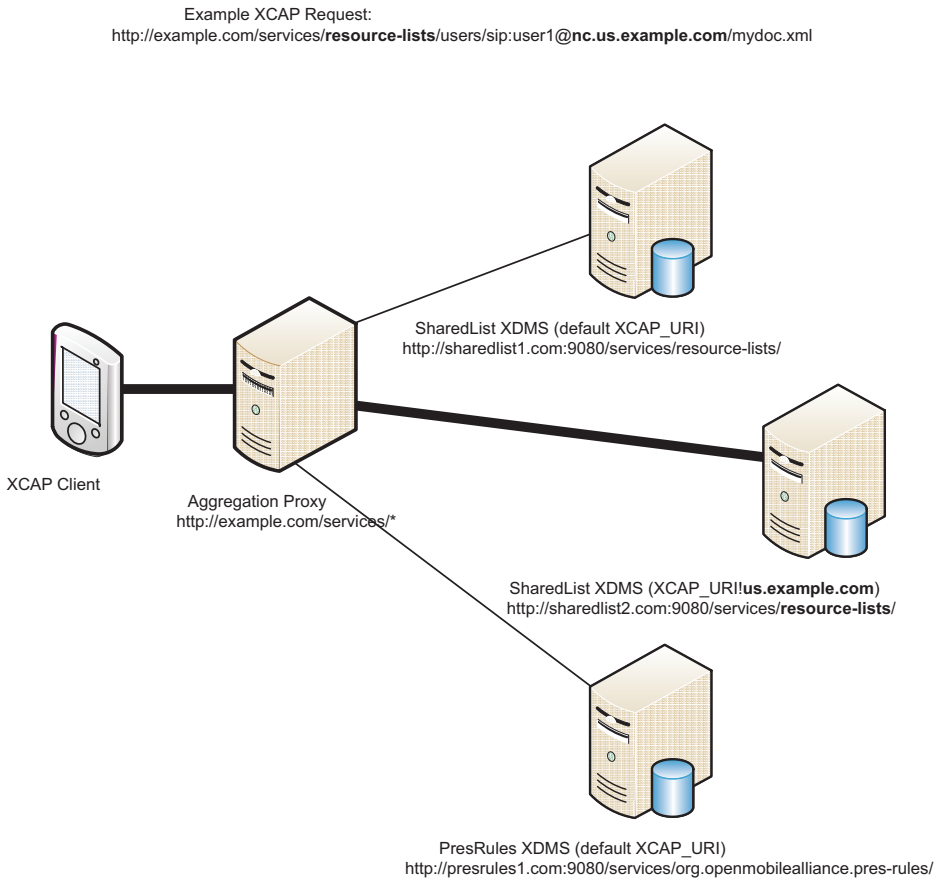


Figure 8. Advanced routing example three

The Aggregation Proxy routes to the `us.example.com` domain because it most closely matches the XUI in the request.

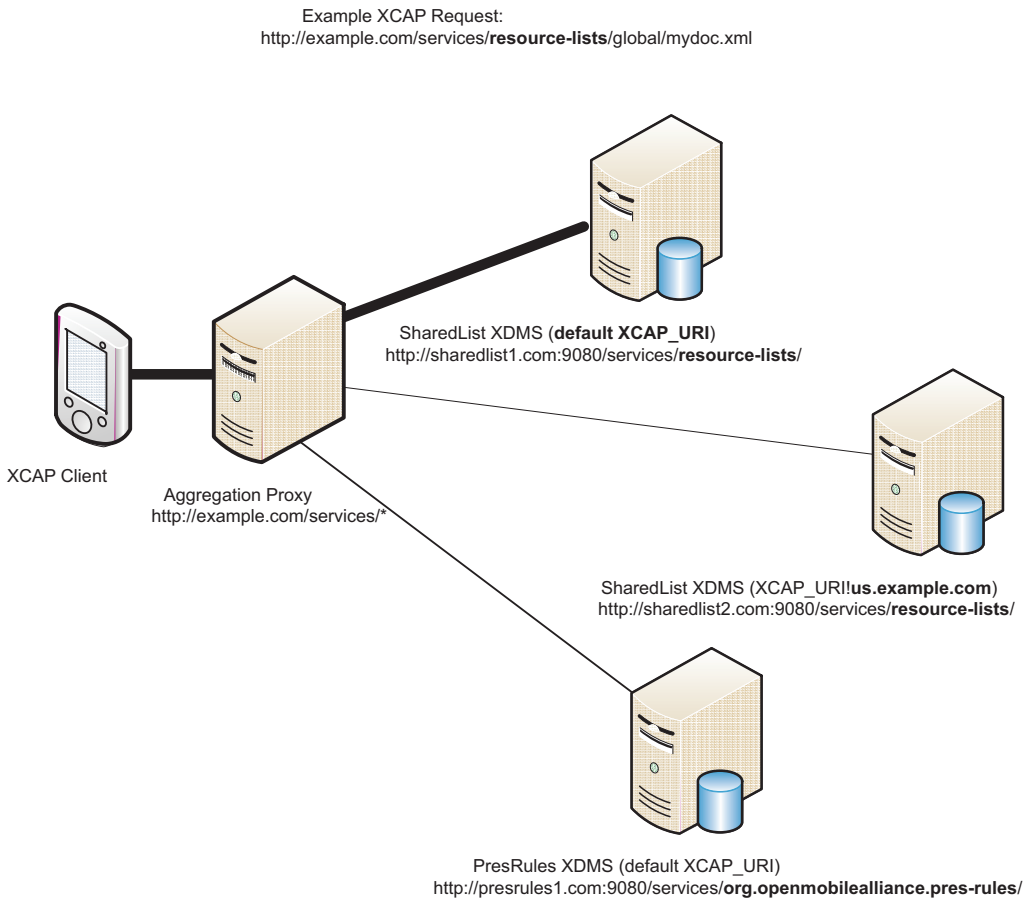


Figure 9. Advance routing example four

The Aggregation Proxy routes all global requests to the default XCAP URI.

Planning for security

To plan for security, consider factors such as requester authentication, requester authorization, mapping, and normalization.

Planning authentication and authorization

The Aggregation Proxy and the IBM XDMS application provide mechanisms for authentication and authorization. (The Trust Association Interceptor (TAI) provides additional security mechanisms as well.)

Requester authentication

The Aggregation Proxy performs HTTP digest authentication using a TAI, which requires the provisioning and configuration of an Lightweight Directory Access Protocol (LDAP) repository containing the subscriber identity and passwords of all users needing access to the system. (In digest authentication, encryption is used so that a user's credentials can be established without the need to transmit a password in plaintext over the network.)

The Aggregation Proxy can also reject authentication attempts after a pre-configured number of failed attempts. This function requires maintaining user session affinity with the initial Aggregation Proxy instance using techniques such as source IP address affinity.

IBM XDMS uses a TAI, which is included with the IBM WebSphere IP Multimedia Subsystem Connector and which enables consumption of the private extension security headers for both HTTP and Session Initiation Protocol (SIP) traffic. This TAI consumes the headers created by the HTTP Digest TAI that runs with the Aggregation Proxy.

If the Aggregation Proxy is not used in the environment and authentication security is still required, then authentication mechanisms can be configured directly on the WebSphere Application Server that is running IBM XDMS. These mechanisms can include a Digest authentication (using a custom TAI), or the built-in WebSphere Application Server global security.

Any IBM-supplied TAI can be replaced with a custom version if the function does not suit your environment.

Requester authorization

For authorization access to user documents, IBM XDMS provides a default behavior that gives each user automatic universal access to all documents that user creates and read access to all global documents. This behavior can be changed through comprehensive or selective provisioning of document authorization documents for affected users.

Confidentiality and integrity

Open Mobile Alliance (OMA) relies on Transport-Level Security (TLS) for ensuring confidentiality on transmitted information and for integrity on information, at least in context of the connection with the previous network element. TLS is supported by WebSphere Application Server.

Note: Security can be turned off when running in a test or pre-production mode. If this mode is used, you will need to configure a default authorization policy in IBM XDMS for anonymous access.

Planning authentication security using the Trust Association Interceptor

The Trust Association Interceptor (TAI) 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 IBM XDMS.

About the scenario

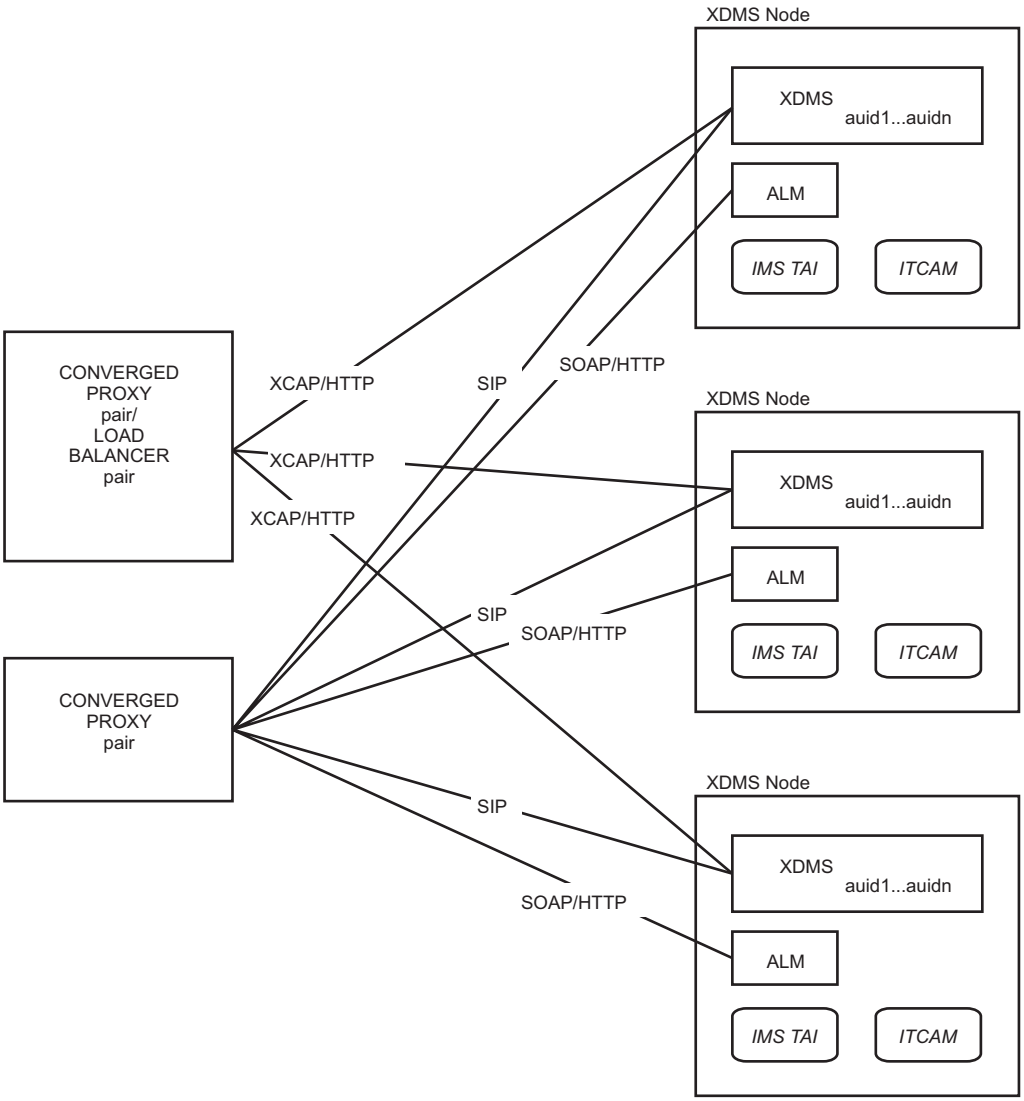
IBM XDMS is an extensible framework that also provides shipped support for two Application Unique ID (AUID) extensions: Shared lists (includes rls-services and resource-lists) and Presence rules (pres-rules). Other AUIDs could be deployed similarly. The constituent components are deployed and scaled separately from each other, so there are different cluster descriptions for the XDMS configuration.

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

The following diagram illustrates the scenario. Three IBM XDMS nodes, with the Trust Association Interceptor deployed on each one, receive SIP traffic that flows through a converged proxy. An Address List Manager on each node receives SOAP/HTTP traffic flowing through the converged proxy. Additionally, each IBM XDMS instance also receives XCAP/HTTP traffic flowing through a different converged proxy (or load balancer) pair.



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, but the same instance could also be used for HTTP traffic.
- The ALM service is a separately-deployed Parlay X Address List management service that is configured to not require TWSS Web service implementations.
- The Trust Association Interceptor detects authenticated user identity from inbound messages.
- IBM Tivoli Composite Application Management (ITCAM) users communicate directly with the IBM XDMS node for purposes of collecting PMI data from that node.
- An aggregation proxy is the reverse proxy security server (RPSS) that performs user authentication before Trust Association Interceptor/XDMS is invoked.
- The aggregation proxy adds an X-XCAP-Asserted-Identity header in the HTTP request.

- The Trust Association Interceptor searches for the asserted identity header.

Normalization and mapping considerations

There are several considerations concerning how normalization and mapping are implemented.

Normalization functions

With IMS, user identifiers (UIDs) are normally represented as uniform resource indicators (URIs). URIs used by SIP may contain a scheme, port, and URI parameters that you may want to strip or remove. In addition, your environment may require that UIDs be case sensitive. Service URIs can also be normalized, service URIs are URIs that identify a service or resource. Service URIs are used as Session Initiation Protocol (SIP) Request URIs and they are identified in rls-services documents.

The normalization properties are defined in the properties file: *was_root*/properties/WebSphere_IMS_Common_Utils.properties.

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

There must be a normalization properties file on each XDMS server with the same properties values. The application server(s) needs to be restarted for changes in the normalization properties file to take effect.

Note: If you change the normalization or mapping properties, you run the risk of orphaning documents that were stored using different mapping properties. These changes should be made in pre-production. If changes are made in production, you may have to recreate the database to remove or recover documents stored with the old mapping properties.

The normalization function has several configurable parameters for UID and service URIs.

Table 2. UID normalization properties

property	description	default value
enableMultipleIDMapping	Strips the schema data from the UID.	`true`
isCaseSensitive	Set to true if the UID is case sensitive. If false, the resulting UID will be converted to all lower case.	`true`
idMappingProvider	Class that implements the Interface for returning a list of matching UIDs or finding a primary UID.	`empty`

Table 3. Service URI normalization properties

property	description	default value
enableMultipleServiceURIMapping	Set to true if the schema should be stripped from a service URI.	`false`
URIParametersFilter	Comma delimited list of parameters to keep attached to Service URI, all others are stripped. Set to star'*' to keep all parameters. If property is empty, all parameters are stripped. Parameters that remain will be sorted in alphabetical order for the resulting Service URI.	`empty`
stripPort	Set to true if the service URI will have the port stripped.	`true`

The following describes how normalization functions.

XML Configuration Access Protocol (XCAP)
The XCAP User Identifier (XUI) in the document selector is always normalized.

SIP The UID of the document parameter in the event header is normalized. If the document is global/index, it finds a list with a service uri= that matches what is in the subscribe header of the SIP SUBSCRIBE. If the document is not specified, it subscribes to the home directory of the user in the To header, that UID will be normalized and the subscription will be to the directory of the normalized UID.

Policy Comparing principal to XUI in the document selector.

Comparing principal to entry in policy document.

Public identity mapping

For the IBM XDMS, the primary purpose of multiple public identity mapping is access control. If users are granted access to a document with one public identity, they can be allowed to access that document with another public identity. The multiple public identity mapping feature enables the mapping of public identities to private ones and has a normalization feature for UIDs and service URIs.

The manner in which public identity mapping is implemented is up to the IBM XDMS service provider. Telecom Application Enablement Feature provides an interface for public identity mapping and the service provider develops the logic that works with their user repository implementation.

Planning to migrate from a previous release of IBM WebSphere XML Document Management Server Component

For users of previous versions of the IBM XDMS product, there are several issues to consider when migrating your existing configuration to the current version.

After you have migrated from the previous version to version 7.0, you will want to test your system to ensure a successful migration. It is recommended that you consider all of the following factors:

- Verify that both versions of the OSGi bundle are running properly on WebSphere Application Server.
- Verify that the 7.0 nodes are updating the databases properly.
- Verify that your Resource Environment Provider (REP) properties are being recognized.
- Verify that the filter classes and helper functions are the same as those used in the previous version.
- Verify that the SIBus notification is functioning properly.
- Verify that your authorization policy documents–which control authorization–are still specifying the correct node selector template. If existing authorization policy documents specify a node selector template that uses one attribute, and the policy has been changed during the upgrade to use another attribute, then all of the authorization policy documents must be updated to conform to the new attribute. (For more information about authorization policy documents, refer to the topic *Policy based authorization*.)

If any of these conditions are not met, then it is a good practice to use dual release clusters. A version 7.0 clone of the existing production cluster is deployed, and then you can reroute traffic from the existing cluster to the new version 7.0 cluster. The document database can be shared between the dual release clusters so that the same documents in the previous cluster can also be managed in the new version 7.0 cluster.

Coexistence with other components

IBM XDMS version 7.0 can interoperate with the following software:

- Aggregation Proxy, version 7.0 or version 6.2
- Shared List application, version 7.0 or version 6.2
- Presence Rules application, version 7.0 or version 6.2
- IBM WebSphere Presence Server Component, version 7.0 or version 6.2
- IBM XDMS, version 7.0 or version 6.2

When you run IBM XDMS, version 7.0 together with IBM XDMS version 6.2 – either in the same cluster or across different clusters – be aware of the following limitations:

- Session Initiation Protocol (SIP) Sessions are not replicated or forwarded from one release cluster to another. Therefore, if you reroute from the previously existing cluster to the new 7.0 cluster, your old subscriptions are lost.
- SIP Notification does not cross release clusters. If there is a subscription to a document in the previously existing cluster but the same document is modified in the new version 7.0 cluster, the subscriber will not be notified because the SIP Notification does not cross over between the clusters and vice versa.
- You may need to double the number of IBM XDMS servers.

Migrating applications

If you used version 6.2 of IBM XDMS to develop applications and you plan to upgrade to version 7.0, contact your IBM representative for advice on migrating the applications.

Chapter 3. Installing

The installation of IBM WebSphere XML Document Management Server Component (IBM XDMS) involves several different file entities, and it requires administrative access to the application server and a database server.

IBM WebSphere XML Document Management Server Component version 7.0 consists of the following installable entitles:

- `com.ibm.ws.xdms_7.0.0.jar`: An OSGi bundle that contains the IBM XDMS core and it must be installed in the plugins directory of each WebSphere installation.
- `IBMSharedListXdms.ear`: An enterprise application that is deployed to support the Open Mobile Alliance (OMA) SharedList IBM XDMS enabler.
- `IBMSharedGroupXdms.ear`: An enterprise application that is deployed to support the OMA SharedGroup IBM XDMS enabler.
- `IBMSharedPolicyXdms.ear`: An enterprise application that is deployed to support the OMA SharedPolicy IBM XDMS enabler.
- `IBMSharedProfileXdms.ear`: An enterprise application that is deployed to support the OMA SharedProfile IBM XDMS enabler.
- `IBMSharedEnablersXdms.ear`: An enterprise application that contains all of the functionality of the other enablers.
- `IBMPresenceRulesXdms.ear`: An enterprise application that is deployed to support the OMA Presence Rules IBM XDMS enabler.
- `IBMXdmsAggregationProxy.ear`: An enterprise application that is deployed to support the OMA Aggregation Proxy service.

Installing IBM XDMS requires administrative access to WebSphere Application Server and to a database server (DB2 or Oracle).

Preparing the WebSphere Application Server environment

You must prepare WebSphere Application Server for the installation of IBM XDMS.

Before you begin

Before installing IBM XDMS, you must have met the following prerequisites:

- WebSphere Application Server Network Deployment version 7.0.0.1 has been installed with a deployment manager with at least one managed node.
- You have federated all nodes into the deployment manager cell.
- A WebSphere user account repository is created. A federated Lightweight Directory Access Protocol (LDAP) repository is recommended. See WebSphere Application Server Information Center for more instructions.
- One of the following database server and client pairs:
  - IBM DB2 Enterprise Server Edition 9.5 FixPak 1 has been installed along with the license for pureXML.
  - The IBM DB2 Enterprise Server Edition 9.5 FixPak 1 client is in the same path on every WebSphere Application Server node in the cluster. This allows you to declare the WebSphere variables for the JDBC driver at the cell level.
  - Oracle Database 11g Enterprise Edition Server.

  - Oracle Database 11g Enterprise Edition Client installed on deployment manager and managed nodes on all servers.
- The Trust Association Interceptor (TAI) is installed and configured. See *Configuring the Trust Association Interceptor* in the IP Multimedia Subsystem Connector portion of the information center for details.

About this task

Unpack the installation tar file in the *was_root* directory.

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`

Complete the following steps to prepare WebSphere Application Server Network Deployment version 7.0.0.1 for the IBM XDMS installation:

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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. 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 the following line of text indicating the version:

      ```
      WebSphere Application Server Network Deployment, 7.0.0.1
      ```
3. Configure security:
   a. Click **Security** → **Global security**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Security** → **Secure administration, applications, and infrastructure**.
   b. Under Administrative security, select **Enable administrative security**. When you enable administrative security, you must configure user repositories. For additional information about user repositories, refer to the WebSphere Application Server Information Center.
   c. Under Application security, select **Enable application security** .
   d. Under Java 2 security, deselect **Use Java 2 security to restrict application access to local resources**.

e. Click **Apply**.
f. Click **Save** to save changes to master configuration.

4. Configure WebSphere Application Server Web security.
   a. In the Global security window, under Authentication, click **Web and SIP security → General settings**.

      Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Web Security → General settings**.
   b. Select **Authenticate only when URI is protected** and the sub heading, **Use available authentication data when an unprotected URI is accessed**.
   c. Click **Apply**.
5. Verify that the console preferences are set to synchronize changes across the nodes:
   a. In the navigation panel, click **System administration → Console preferences**.
   b. Select **Synchronize changes with Nodes**.
   c. Click **Apply**.
   d. Click **Save** to save changes to the master configuration.
6. Stop all application servers, node agents, and the deployment manager.
   a. Stop all application servers.
   b. Stop all node agents.
   c. Stop the deployment manager.

   The servers will be restarted in a subsequent section.

What to do next

The clusters can now be created.

Creating XDMS clusters

Before installing applications, you can create a cluster and add cluster members for each enabler using the Integrated Solutions Console.

Before you begin

Before installing each application, you must have the following software installed:

WebSphere Application Server Network Deployment, version 7.0.0.1

In addition, you should have:

- Created a deployment manager profile
- Federated all nodes into the deployment manager cell
- Prepared the environment

About this task

Create a cluster for **each** enabler, and add members by completing the following steps:

Note: The enablers you are able to install are: Shared Profile, Shared Policy, Shared Group, Shared List, and Presence Rules (Shared List and Presence Rules are required for use with IBM WebSphere Presence Server).

1. Start the deployment manager.
2. Start the node agent.
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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
4. 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. Click **New**.
6. Type the cluster name, for example `SharedListCluster`.
7. Select **Prefer local**.
8. Select **Configure HTTP session memory-to-memory replication**.
9. Click **Next**.
10. Type *server_name* in the **Member name** field. This is the fully qualified host name of additional WebSphere Application Server servers being added as cluster members – for example, `SharedList0`.
11. Select **Generate unique HTTP ports**.
12. Select the appropriate option for **Select basis for first cluster member** for your environment. For example, select **Create the member using an application server template** to create a new application server using the existing default template.
13. Click **Next**.
14. Optional: If your environment requires more than one cluster member, add additional cluster members. Repeat the following steps for each cluster member you would like to add.
    a. Type *server_name* in the **Member name** field. This is the fully-qualified host name of additional WebSphere Application Server servers being added as cluster members – for example, `SharedList1`.
    b. Select the node for the cluster member.
    c. Select **Generate unique HTTP ports**.
    d. Click **Add Member** to add the cluster member.
15. Click **Next**, and click **Finish**.
16. Click **Save** to save changes to the master configuration.
17. Click **OK**.

What to do next

You must create a proxy server and associate the proxy server with the cluster that you have created. Because IBM XDMS supports both Session Initiation Protocol (SIP) and HTTP, you will need a WebSphere Application Server proxy server that supports HTTP and SIP requests. Refer to the WebSphere Application Server Information Center for additional information on setting up the proxy server.

Note: Once you have created the proxy server you need to disable page caching for the HTTP proxy.

Creating the Aggregation Proxy cluster

Create a cluster for the Aggregation Proxy in the WebSphere Application Server Network Deployment console, and use the default values that WebSphere Application Server provides.

Before you begin

Before you begin, the following software should be installed:

WebSphere Application Server Network Deployment, version 7.0.0.1

In addition, you should have:

- Created a deployment manager profile
- Federated all nodes into the deployment manager cell
- Configured security

About this task

Create the AggProxyCluster cluster, and add members, 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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. 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**.
3. Click **New**.
4. Type `AggProxyCluster` for the **Cluster name**.
5. Select **Prefer local**.

6. Click **Next**.
7. Type the **Member name**. This is the fully qualified host name of additional WebSphere Application Server servers being added as cluster members.
8. Select **Generate unique HTTP ports**.
9. Select the appropriate option for **Select basis for first cluster member** for your environment.
   - Select **Create the member using an application server template** to create a new application server using an existing template.
   - Select **Create the member using an existing application server as a template** to create a new application server using another application server as a template.
   - Select **Create the member by converting an existing application server** to select an application server from a federated node as the first cluster member.
10. Click **Next**.
11. Optional: If your environment requires more than one cluster member, add additional cluster members. Repeat the following steps for each cluster member you would like to add.
    a. Type *server_name* the member you would like to create. For federated nodes with multiple cluster members, you will need to know the names of each of the members.
    b. Select the node for the cluster member.
    c. Select **Generate unique HTTP ports**.
    d. Click **Add member** to add the cluster member.
12. Click **Next**.
13. Click **Finish**.
14. Click **Save** to save changes to the master configuration.
15. Click **OK**.

What to do next

You must create a proxy server and associate the proxy server with the cluster that you have created. Because IBM XDMS supports both Session Initiation Protocol (SIP) and HTTP, you will need a WebSphere Application Server proxy server that supports HTTP and SIP requests. Refer to the WebSphere Application Server Information Center for additional information on setting up the proxy server.

Note: Once you have created the proxy server you need to disable page caching for the HTTP proxy.

Configuring ports

The HTTP transport ports and virtual hosts must be properly defined for the clusters. You can verify and correct the port configuration using the Integrated Solutions Console.

Before you begin

Before you begin, the following software should be installed:

- WebSphere Application Server Network Deployment, Version 7.0.0.1

In addition, you should have:

- Completed the environment preparation steps
- Created the cluster you wish to install the IBM XDMS EAR files on

About this task

To ensure that the ports are configured correctly for **each** enabler, complete the following steps:

Note: The enablers you are able to install are: Shared Profile, Shared Policy, Shared Group, Shared List, and Presence Rules (Shared List and Presence Rules are required for use with IBM WebSphere 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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. Verify that the HTTP transport ports are properly defined.
   a. 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 ***server_name*** for the cluster member you want to verify.
   c. On the right side of the page, in the Communication section, Expand **Ports**.
   d. Verify that the correct ports are listed. Two ports must be included in the virtual host definition:
      - WC_defaulthost (typically 9080)
      - SIP_DEFAULTHOST (typically 5060)

      Note: You need to remember the port numbers, because in the next step you will verify that the ports are correct for each virtual host.
3. Verify that the virtual hosts are properly defined.
   a. In the navigation panel, click **Environment** → **Virtual Hosts**.
   b. Click the ***virtual_host_name*** that you are using. Typically, this is `default_host`.
   c. Under Additional Properties, click **Host Aliases**.
   d. Verify that all ports are correct.
4. If the information is not correct, make necessary changes.
   a. To change an existing port, click ***host_name*** and modify as necessary.
   b. To add a missing port, click **New**. Supply values for **Host Name** and **Port**, and click **OK**.
   c. Click **Apply**.

d. Click **Save** to save changes to the master configuration.
e. Click **OK**.

Creating authentication users and groups

Using the deployment manager profile, prepare for the installation by creating users and groups with the appropriate privileges.

Before you begin

Before you begin, the following steps should be completed:

- A WebSphere user account repository is created. A federated LDAP repository is recommended. See WebSphere Application Server Information Center for more instructions.
- Started the application server.
- Connected to the database.

About this task

Follow these steps to create your users and groups.

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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. Click **Users and Groups → Manage Users**.
3. Create a user with super administrative privileges.
   a. Click **Create**.
   b. Type a **User ID**. For example, type `superadmin`.
   c. Type the **First name** and **Last name** for the user.
   d. Optional: Type the **E-mail** for the user. Use the standard e-mail format. For example, `user@example.com`
   e. Type the **Password** for the user. For example `superadmin`, password `superadminpassword`
   f. Confirm the password.
   g. Click **Create**. The following success message displays: `The user was created successfully.`
   h. Click **Close**.
   i. In the navigation panel, click **Users and Groups → Administrative User Roles**.
   j. Click **Add**.

k. Type the **User ID** for the user you created. For example, type `superadmin`.
l. Press Ctrl and click **Administrator** and **Configurator** for the **Roles** of the user.
m. Click **OK**.
n. Click **Save** to save to the master configuration

4. Create an anonymous user to use IBM XDMS.
   a. Click **Create**.
   b. Type a User ID. For example, type `anonymous.invalid`
   c. Type the First name and Last name for the user.
   d. Optional: Type the E-mail for the user. Use the standard e-mail format. For example, user@example.com
   e. Type the password for the user.
   f. Confirm the password.
   g. Click **Create**. The following success message displays: `The user was created successfully.`
   h. Click **Close**.

Configuring the Oracle thick client

If you are using Oracle Database, install the Oracle thick client and configure it to work with IBM XDMS.

About this task

You can configure a thick client or a thin client for Oracle. To configure the thin client, you will need to install XDMS 7.0 interim fix 1. Contact your IBM representative for more information.

If you elect to use the thin client, you do not need to install the Oracle on any server where WebSphere Application Server is running. You can disregard this procedure.

However, if you do not plan to install XDMS 7.0 interim fix 1 immediately, you must continue with the steps for installing and configuring the Oracle thick client. Install the client on the deployment manager and on all managed nodes.

In this procedure, you will set up the Oracle thick client to use the `tnsnames.ora` file.

In the following steps, *oracle_home* is the directory in which the Oracle thick client is installed.

1. Open a command prompt and type the following command:
   `mkdir -p` *`oracle_home`*`/network/admin`
2. Retrieve the `tnsnames.ora` file from the database server with which the Oracle thick client will communicate.
3. Put the `tnsnames.ora` file into the *`oracle_home`*`/network/admin` directory on the Oracle client.
4. Change (`cd`) to the following directory: *`was_root`*`/profiles/`*`app_server_name`*`/bin`.
5. Edit the `setupCmdLine.sh` file as follows:

a. Add the following environment variables at the end of the file:

Table 4. Environment Variables

Variable	Description	Example
ORACLE_HOME=	Path where the Oracle client is installed	`/opt/app/oracle/product/11.1.0/client_1`
TNS_ADMIN=	TNS Admin directory	`$oracle_home/network/admin`
CLASSPATH=	Class path	For WebSphere 7.0.0.1: `$oracle_home/ojdbc6.jar` For WebSphere 6.1.0.21: `$oracle_home/ojdbc5.jar`
LIBPATH=	Library path (normally used for AIX)	$LIBPATH:/usr/lib:/lib:/usr/local/lib:$*oracle_home*
LD_LIBRARY_PATH=	Library path (normally used for Linux)	$LD_LIBRARY_PATH:$*oracle_home*/network/lib:$*oracle_home*/lib:$*oracle_home*

b. Add the following `export` command at the end of the file:

```
export ORACLE_HOME TNS_ADMIN CLASSPATH LIBPATH LD_LIBRARY_PATH
```

6. Save your changes and exit the editor.
7. Restart the application server and the node agent.

Preparing the Trust Association Interceptor for use

Follow these procedures to install the Trust Association Interceptor (TAI) for use in conjunction with the IBM WebSphere XML Document Management Server Component.

Later, the IBM XDMS installer will perform additional customization to make the TAI ready for use.

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 TAI can be used with IBM XDMS.

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

IBM XDMS requires databases for storing XML documents managed by IBM XDMS, for the service integration bus, and for storing usage records.

Preparing the DB2 databases

To prepare DB2 and create the required databases, complete these procedures.

Creating the DB2 database and database tables for the XDMS enablers

Create the DB2 database for the enabler applications that you plan to deploy.

Before you begin

Before you begin, you must have completed the following task:

- Installed and started IBM DB2 Enterprise Server Edition, 9.5 FixPak 1
- Installed the DB2 client or copied the DB2 client files to all WebSphere servers

About this task

Copy `IBMXdmsDbPackage_7.0.0.tar` from the installation media and untar it on your system. Then, switch to the XDMS directory of the `IBMXdmsDbPackage_7.0.0.tar`. A database-creation script is provided for each enabler.

The scripts are named as follows:

Shared List XDMS: `IBMSharedListXdmsDb2.sh`

Presence Rules XDMS: `IBMPresenceRulesXdmsDb2.sh`

Shared Profile XDMS: `IBMSharedProfileXdmsDb2.sh`

Shared Policy XDMS: `IBMSharedPolicyXdmsDb2.sh`

Shared Group XDMS: `IBMSharedGroupXdmsDb2.sh`

Shared Enablers XDMS: `IBMSharedEnablersXdmsDb2.sh`

Create only the databases that you need.

1. Copy the scripts from one of the WebSphere Application Server nodes to a location on the database server that can be accessed by the instance owner.
2. Log in to DB2 as the DB2 instance owner, for example *db2inst1*.
3. Start the DB2 instance.
4. Make the scripts executable by executing the following command for *each* script that you plan to use: `chmod 755 script_name.sh`
5. Create the databases and database tables by running the following command for each enabler that you plan to deploy:

   ```
   ./script_name.sh db_name db_user db_password
   ```

   For example:

   ```
   ./IBMSharedListXdmsDb2.sh XdmsSL db2inst1 password
   ```

6. Verify that the XDMS databases were created properly by executing the following command:

   ```
   db2 connect to db_name user db_user using db_password
   ```

   For example:

   ```
   db2 connect to XdmsSL user db2inst1 using password
   ```

   You should see results that are similar to the following:

```
Database Connection Information

 Database server        = DB2/LINUX 9.5.1
 SQL authorization ID   = DB2INST1
 Local database alias   = db_name
```

7. Verify that all tables were properly created by executing the `db2 list tables` command.

   The display should show one table for each AUID represented by the enablers that you plan to deploy. AUIDs for the enablers are listed under the topic *XDMS enablers*.

8. Retain the following information. You will need it when you install and configure each enabler and when you connect to the databases for the enablers.
   - Database name
   - Connection port
   - DB2 instance server hostname
   - DB2 instance userid and password

Creating the Service Integration Bus DB2 database

Create the Service Integration Bus database.

Before you begin

Before you begin, you must have completed the following task:

- Installed and started IBM DB2 Enterprise Server Edition, 9.5 FixPak 1
- Installed the DB2 client or copied the DB2 client files to all WebSphere servers

About this task

Copy `IBMXdmsDbPackage_7.0.0.tar` from the installation media and untar it on your system. Then, switch to the XDMS directory of the `IBMXdmsDbPackage_7.0.0.tar`. A database-creation script is provided for each enabler.

1. Log in to DB2 as the DB2 instance owner, for example *db2inst1*.
2. Run the following command:

   ```
   ./createXdmsSIBusDb2.sh db_name db_user db_password
   ```

   For example:

   ```
   ./createXdmsSIBusDb2.sh XDMSSIB db2inst1 password
   ```

3. Run the following command to verify that the XDMSSIB database was created properly:

   ```
   db2 connect to db_name user db_user using db_password
   ```

   For example:

   ```
   db2 connect to XDMSSIB user db2inst1 using password
   ```

   You should see the following results:

   ```
   Database Connection Information

    Database server        = DB2/LINUX 9.1.2
    SQL authorization ID   = DB2INST1
    Local database alias   = XDMSSIB
   ```

4. Retain the following information, which you will need to install and configure XDMSSIB and connect to the database.
   - Database name

- Connection port
- DB2 server hostname
- DB2 userid and password

Creating the DB2 usage record database

Create the DB2 usage record database.

Before you begin

Before you begin, you must have completed the following task:

- Installed and started IBM DB2 Enterprise Server Edition, 9.5 FixPak 1
- Installed the DB2 client or copied the DB2 client files to all WebSphere servers

About this task

Complete the following steps on the database server where you want to create the database. The following instructions use the interactive menu. You can run the script as a single command. Additional information about using the single command is provided in the script.

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, IBMXdmsDbPackage_7.0.0.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 IBMXdmsDbPackage_7.0.0.tar
   ```

6. Switch to the following directory: *DB_dir*`/usageRecords`.
7. Run the crtSrvDb2.sh script using the command: ./crtsrvDb2.sh *dbServer dbPort dbLocal dbNodeName dbName dbAlias dbLocale dbAdmin dbAdminPW dbUser dbUserPW INITorDDLFile infoflag*, for example:

   ```
   ./crtsrvDb2.sh dh_hostname 50000 TRUE RDBSRV XDMSURDB XDMSUR US db2inst1 password db2inst1 passwo
   ```

   Note: This command must be run twice. Once with the INITorDDLFile set to INIT and a second time with the same parameters set to UsageDbDb2.ddl.

   The following table provides the parameter information for the command:

Parameter	Description	Example value
dbServer	Database server hostname	Fully qualified host name
dbPort	Database server connection port	50000
dbLocal	Local database	TRUE for local
dbNodeName	Remote database server node name	RDBSRV
dbName	Database name	XDMSURDB
dbAlias	Database alias name	XDMSUR
dbLocale	Database territory code	US
dbAdmin	Database server administrator instance ID	db2inst1

Parameter	Description	Example value
dbAdminPW	Database server administrator instance password	*pw*
dbUser	Database user ID	db2inst1
dbUserPW	Database user password	*pw*
INITorDDLFile	'INIT' to initialize or the path to DDL file	UsageDbDb2.ddl
infoflag	Display information flag	TRUE

8. Verify the configuration tables were created properly.
9. Verify the tables were created properly:
   a. Connect to the database using the following DB2 command:
      `db2 connect to XDMSUR user user_name using password`
   b. Verify the tables were created using the following DB2 command:
      `db2 list tables for schema user_name`

      For example, if the user name is db2admin, use the `db2 list tables for schema db2admin` command.

      The following tables should be listed:
      - USAGERECORDS

Preparing the Oracle databases

To prepare Oracle Database and create the required databases, complete these procedures.

You can configure a thick client or a thin client for Oracle. To configure the thin client, you will need to install XDMS 7.0 interim fix 1. Contact your IBM representative for more information.

Creating Oracle database tables for the XDMS enablers

You must create Oracle databases and database tables for the enabler applications that you plan to deploy.

Before you begin

Before you begin, the following software should be installed:

- Oracle Database, version 11.1.0.7

After you have unpacked the `IBMXdmsDbPackage_7.0.0.tar` file, follow the Oracle procedures to create databases for each of the following enablers that you plan to deploy:

- Shared List XDMS
- Presence Rules XDMS
- Shared Profile XDMS
- Shared Policy XDMS
- Shared Group XDMS
- Shared Enablers XDMS

About this task

A script for creating database tables is provided for each enabler. The scripts are located in the `installableApps` directory for the installed WebSphere Application Server instance–for example, *was_root*`/installableApps/xdms/scripts/dbScripts/xdms/`.

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`

The scripts are named as follows:

Shared List XDMS: `IBMSharedListXdmsOracle.sh`

Presence Rules XDMS: `IBMPresenceRulesXdmsOracle.sh`

Shared Profile XDMS: `IBMSharedProfileXdmsOracle.sh`

Shared Policy XDMS: `IBMSharedPolicyXdmsOracle.sh`

Shared Group XDMS: `IBMSharedGroupXdmsOracle.sh`

Shared Enablers XDMS: `IBMSharedEnablersXdmsOracle.sh`

Create only the database tables that you need.

1. Copy the scripts from one of the WebSphere Application Server nodes to a location on the database server that can be accessed by the instance owner.
2. Log in to Oracle as the instance owner (system).
3. Start the Oracle instance.
4. Make the scripts executable by executing the following command for *each* script that you plan to use: `chmod 755 script_name.sh`
5. Create the database tables by running the following command for each enabler that you plan to deploy: `./script_name.sh db_name db_user db_password` For example:

   ```
   ./IBMSharedListXdmsOracle.sh XdmsSL db2inst1 password
   ```

6. Use the Oracle SELECT command to verify that all tables were properly created. (Refer to the Oracle documentation for details about the command syntax.)

   The display should show one table for each AUID represented by the enablers that you plan to deploy. AUIDs for the enablers are listed under the topic *XDMS enablers*.
7. Retain the following information. You will need it when you install and configure each enabler and when you connect to the databases for the enablers.
   - Database name
   - Connection port
   - Oracle instance server hostname
   - Oracle instance user ID and password

Creating the usage records database and tables for Oracle

You must configure an Oracle usage records database. The `crtsrvOra.sh` script is provided for this purpose.

Before you begin

Before you begin, the following software should be installed:

- Oracle Database, version 11.1.0.7

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, `IBMXdmsDbPackage_7.0.0.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 IBMXdmsDbPackage_7.0.0.tar
   ```
6. Switch to the following directory: *DB_dir*`/usageRecords`.
7. Delete the usage record table, if it has already been created.
8. Run the `crtsrvOra.sh` script using the following command
   ```
   ./crtsrvOra.sh dbName dbAdmin dbAdminPW dbUser dbUserPW INITorDDLFile infoflag
   ```

   Use the following parameter values:

Parameter	Description	Example value
dbName	Database name	XDMS
dbAdmin	Database administrator ID	SYSTEM
dbAdminPW	Database administrator password	*pw*
dbUser	Database user ID	xdmsuser
dbUserPW	Database user password	*pw*
INIT or DDLFile	'INIT' to initialize or path to DDL file	UsageDbOra.ddl
infoflag	Display information flag	TRUE

   Example:
   ```
   ./crtsrvOra.sh XDMSUR SYSTEM password xdmsuser password UsageDbORa.ddl TRUE
   ```
9. After the script displays the input values, press Enter to continue.
10. After the script completes, verify that the database tables were created properly by running a command similar to the following:
    ```
    select USAGERECORDS from user_tables
    ```

    If the script ran properly, the USAGERECORDS table is shown in the display.

Configuring IBM XDMS to work with Oracle Database

You must configure IBM XDMS so that it will work with an Oracle thick client.

About this task

As an alternative to installing the thick client, you can configure a thin client for Oracle Database. To configure the thin client, you will need to install XDMS 7.0 interim fix 1. Contact your IBM representative for more information.

Follow these steps to configure IBM XDMS to work with an Oracle thick client.

1. Make two directories for the Oracle XML files, for example `/usr/Oracle_XML_files` and `/usr/Oracle_jdbc_files`.

2. Copy the following files to the `Oracle_jdbc_files` directory:

   `ojdbc6.jar` (for WebSphere Application Server version 7.0.0.1)

   `ojdbc5.jar` (for WebSphere Application Server version 6.1.0.x)

   Copy the following files to the `Oracle_XML_files` directory:

   `xdb.jar`

   `xmlparserv2.jar`

   Note: The ojdbc* JAR file must be in a different directory than the xdb and xmlparserv2 JAR files. If the xdb and xmlparserv2 JAR files are in the directory that is pointed to by the data source, the deployment manager could fail to start. If you will be installing XDMS 7.0 interim fix 1 and moving to the thin client, enter the `Oracle_jdbc_files` directory into the Data Source Creation screen of the installer, or else enter the Oracle client installation directory.
3. Add the absolute path to these JAR files in each XDMS Application server JVM:
   a. Open the Integrated Solutions Console.
   b. 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**.
   c. Click **server_name** → **Process Definition** → **Java Virtual Machine** and select the Configuration tab.
   d. Add the absolute path to the JAR files in the **Class Path** text box. Example: `/usr/Oracle_jdbc_files/ojdbc6.jar`, `/usr/Oracle_XML_files/xdb.jar`, or `/usr/Oracle_XML_files/xmlparserv2.jar`.
4. Add the following generic JVM arguments:

```
-Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.DocumentBuilderFactoryImpl
-Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl
-Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryImpl
```

5. Restart the deployment manager and all managed node servers.

What to do next

Repeat these steps on each managed node. Then perform a test connection to make sure that the configuration is successful.

Installing XDMS

Use the interactive installer or silent installation scripts to install the IBM XDMS product. The following topics describe installation and migration procedures.

Upgrading to IBM XDMS version 7.0

To upgrade an existing IBM XDMS version 6.2 to version 7.0, run the installation program. The installation program detects the presence of the existing system and performs the necessary upgrades.

Before you begin

You must have met the following requirements:

- WebSphere Application Server Network Deployment version 7.0.0.1 or 6.1.0.21 is installed with a deployment manager and at least one managed node.
- You have federated all nodes into the deployment manager cell.

- A WebSphere user account repository is created. A federated LDAP repository is recommended. See the WebSphere Application Server 7.0 Information Center for more instructions.
- IBM DB2 Enterprise Server Edition 9.5 FixPak 1 has been installed along with the license for pureXML.
- The IBM DB2 Enterprise Server Edition 9.5 FixPak 1 client is in the same path on every WebSphere Application Server node in the cluster. This allows the WebSphere variables for the JDBC driver to be declared at the Cell level.
- Clusters are created for each IBM XDMS application you plan to install.

About this task

To install IBM XDMS and the XDMS enterprise applications, perform the following steps at the deployment manager node

1. Change directories (`cd`) to the CD-ROM drive.
2. Issue the following command to launch the interactive installer:
   ```
   ./setup
   ```
3. Select **I accept the terms of the License Agreement**.
4. 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.
5. Click **Next**.
6. Select a WebSphere Application Server installation.
7. Click **Next**. The installer looks for an installation of the OSGi bundle for IBM XDMS version 7.0. If none are found, the installation configuration screen displays with the **XDMS Core Files** selected.
8. Click **Next**.
9. Select a WebSphere Application Server installation.
10. Click **Next**.
11. Enter your WebSphere Application Server security information.
    a. Type your WebSphere Application Server fully qualified host name.
    b. Type the SOAP port address for your WebSphere Application Server.
    c. If security is enabled, select **The WebSphere Environment is secure**.
    d. Type an administrative user ID.
    e. Type the password for the administrative user ID.
12. Click **Next**.
13. Select the enablers you are installing from the list of available enablers. For each enabler you select the installer presents a scope selection screen where you must select the proper cluster or server for the application to be installed to. These instructions assume you have selected Shared List and Presence Rules for installation in a clustered environment.
14. Click **Next**.
15. Select the target scope for Shared List, for example **WasScope[type]:ServerCluster[cluster]:SharedListCluster**.
16. Click **Next**.
17. Select the target scope for Presence Rules, for example **WasScope[type]:ServerCluster[cluster]:PresenceRulesCluster**.
18. Click **Next**. The Pre-Installation summary panel appears.
19. Click **Install**.

20. Perform these installation steps again on any remote nodes, installing only the XDMS core files.

    Note: If there are multiple nodes on the same machine as the deployment manager, you *do not* need to rerun the installer on that machine.

21. Start the servers for the installed application.

    Note: During the startup, you can safely ignore any error messages that refer to problems with the UsageRecordWriter.properties file.

Installing using the interactive installer

Launch the interactive installation to install XDMS and desired components.

Before you begin

You must have met the following requirements:

- WebSphere Application Server Network Deployment Version 7.0.0.1 or 6.1.0.21 is installed with a deployment manager and at least one managed node.
- You have federated all nodes into the deployment manager cell.
- A WebSphere user account repository is created. A federated LDAP repository is recommended. See WebSphere Application Server Information Center for more instructions.
- IBM DB2 Enterprise Server Edition version 9.5 FixPak 1 has been installed along with the license for pureXML.
- IBM DB2 Enterprise Server Edition 9.5 FixPak 1 client in the same path on every WebSphere Application Server node in the cluster. This allows the WebSphere variables for the JDBC driver to be declared at the Cell level.
- Oracle Database version 11.1.0.7 has been installed.
- Oracle Database version 11.1.0.7 client in the same path on every WebSphere Application Server node in the cluster. This allows the WebSphere variables for the JDBC driver to be declared at the Cell level.
- Created the clusters for each IBM XDMS application you plan to install.

About this task

To install IBM XDMS and the XDMS enterprise applications, perform the following steps at the deployment manager node:

1. Insert the XDMS installation CD into the CD-ROM drive.
2. Change directories (`cd`) to the CD-ROM drive.
3. Set the proper file permissions on the installer by entering the following from the command prompt:

   ```
   chmod 755 setup
   ```

4. Issue the following command to launch the interactive installer:

   ```
   ./setup
   ```

5. In the Introduction pane, click **Next**.
6. Select **I accept the terms of the License Agreement**.
7. 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.
8. Select a WebSphere Application Server installation.
9. Click **Next**.

10. Select the machine profile for the machine where you are installing. For machines hosting only managed nodes, you should only install the IBM XDMS core files and not the component applications.
11. Click **Next**.
12. Enter your WebSphere Application Server host and security information.
    a. Type the SOAP port address for your WebSphere Application Server.
    b. If security is enabled, select **The WebSphere Environment is secure**.
    c. Type an administrative user ID.
    d. Type the password for the administrative user ID.
13. Click **Next**. The installer looks for an installation of the OSGi bundle for IBM XDMS version 7.0. If none are found, the installation configuration screen displays with the **XDMS Core Files** selected.
14. Optional: You may choose additional components to install. Additional components include:
    - Shared List
    - Presence Rules
    - Aggregation Proxy
    - Shared Policy
    - Shared Group
    - Shared Profile

    Note: The process described in this document will be repeated for each selected component.
15. Click **Next**. The installation configuration is the next screen.
16. Configure the installation:
    a. Select the target scope for the component, for example **WasScope[type]:ServerCluster[cluster]:SharedListCluster** for the Shared List component.
    b. Click **Next**.
    c. Type the required information in the fields for the component you selected.

Table 5. Resource Environment Provider values

Field	Value
XCAP Root	The XCAP Root of the Aggregation Proxy, if installed, or the XCAP Root of the local server (recommended value: http(s)://hostname:ports/services)
Super User User	A user configured with the super-admin role (recommended value: superadmin)
Super User Password	Password for the super-admin role (recommended value: none)

    d. Click **Next**.
    e. Select the database type you are using:

IBM DB2

Oracle

    f. Type the required information in the database fields for the component you selected.

Table 6. Enabler data source information

Field	Value
Database Driver Directory	*db_client_root*/java/ **Note:** Where *db_client_root* is the path where your database client has been installed.
Database Server Host	The host name of your database server, for example *dbServerhostname*.
Database Port	The database port number designated for client communications, for example *50001*.
Database Name or **Oracle SID**	The database name for the XDMS, for example *xdms*.
Database User	The administrative user name for the database instance.
Database Password	The password for the database administrative user.

g. Click **Next**.

h. Select a storage type for Service Integration Bus data: **Data Store** or **Files Store**.

17. Click **Next**. If you have selected another component, that component's installation configuration appears next. Repeat the steps for each component.

    Note: If you selected to install multiple enablers, the common configuration information previously entered for the first one is retained. As each panel is presented, you may change the information as needed.

18. Configure the Service Integration Bus data source.

    Note: This data source will be common to all installed enablers.

    a. Select the database type you are using: **Data Store** or **Files Store**.

    b. Type the required information in the database fields:

Table 7. Service Integration Buss enabler data source information

Field	Value
Database Driver Directory	*db_client_root*/java/ **Note:** Where *db_client_root* is the path where your database client has been installed.
Database Server Host	The host name of your database server, for example *dbServerhostname*. **Note:** For Oracle users this field is not needed and is grayed out.
Database Port	The database port number designated for client communications, for example *50001*. **Note:** For Oracle users, this field is not needed and is grayed out.
Database Name or **Oracle SID**	The database name for the XDMS. This should be *xdmssib*.
Database User	The administrative user name for the database instance.
Database Password	The password for the database administrative user.

19. Click **Next**.
20. Optional: Select **Yes** to configure a usage record database.
    a. Click **Next**.
    b. Select the database type you are using:

       IBM DB2

       Oracle

    c. Type the required information in the database fields:

Table 8. Usage Record enabler data source information

Field	Value
Database Driver Directory	*db_client_root*/java/ **Note:** Where *db_client_root* is the path where your database client has been installed.
Database Server Host	The host name of your database server, for example, *dbServerhostname*. **Note:** For Oracle users this field is not needed and is grayed out.
Database Port	The database port number designated for client communications. For example *50001*. **Note:** For Oracle users, this field is not needed and is grayed out.
Database Name or **Oracle SID**	The database name for the XDMS. This should be *xdmsurdb*.
Database User	The administrative user name for the database instance.
Database Password	The password for the database administrative user.

21. Click **Next**. The Pre-Installation summary panel appears.
22. Click **Install**.
23. Perform these installation steps again on any remote nodes, installing only the XDMS core files.

    Note: If there are multiple nodes on the same machine as the deployment manager, you *do not* need to rerun the installer on that machine.
24. Start the servers for the installed application.

    Note: During the startup, you can safely ignore any error messages that refer to problems with the UsageRecordWriter.properties file.

Silent installation

The IBM XDMS program code can be installed silently without user interaction, by reading user responses from a response file.

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 you begin the silent installation.

About this task

Customizing the response file 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: *was_root*/logs/XDMS/xdmsTraceInstall.log.

To edit your response file, perform the following procedure on the deployment manager node.

1. Navigate to *was_root*/installableApps/xdms/samples.

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

   - /usr/IBM/WebSphere/AppServer
   - /opt/IBM/WebSphere/AppServer

2. Open the installation response file, xdms_install.rsp, in a text editor.
3. Edit the response file, modifying the following values as needed. The following table lists the response file parameters that are used for installation.

Table 9. response file properties

response section	property	valid values	example value
License			
	LICENSE_ACCEPTED=	Must equal true to use installer	*true*
WebSphere Variables			
	WAS_HOME=	valid path on the local machine with no trailing slash	*/opt/IBM/WebSphere/AppServer*
	WAS_HOST=	Valid hostname or IP address	*localhost*
	WAS_PORT=	Valid port number	*8879*
	WAS_SECURED=	0 for false, 1 for true	*1*
	WAS_USER=	Valid WebSphere Application Server administrative user name	*was_admin*
	WAS_PASSWORD=	Valid WebSphere Application Server administrative user's password	*waspass*
	REMOTE_NODE	Is this maintenance being performed on a standalone machine (0), deployment manager (0), or managed node (1)	*0* or *1*

Table 9. response file properties (continued)

response section	property	valid values	example value
	PROFILE_HOME	If this maintenance is being performed on the deployment manager machine, or a machine with a standalone node, then you must specify the Dmgr or node's profile home directory	*/opt/IBM/WebSphere/AppServer/profiles/Dmgr01*
Installer Actions			
	CORE_FILES_ACTION=	Action to take for IBM XDMS program code: INSTALL, UNINSTALL, UPGRADE, or NONE	*INSTALL*
	ENABLER_ACTION	Action to take for the XDMS enablement applications: INSTALL, UNINSTALL, UPGRADE, or NONE	*INSTALL*
	ENABLER_ACTION _ARGS=	For Install action: • -ear *path_to_ear* • -cluster *WAS_cluster_name* • -server *WAS_server_name* • -node *WAS_node_name* For Uninstall action: • -app *WAS_app_name* For Upgrade action: • -app *WAS_app_name* • -ear *path_to_ear*	For Install action: • -ear */dir/IBMSharedListXdms.ear* • -cluster *SharedListCluster* • -server *SharedListCluster0* • -node *SharedListNode* For Uninstall action: • -app *Shared List XDMS* For Upgrade action: • -app *Shared List XDMS* • -ear */path/IBMSharedListXdms.ear*
	ENABLER_SCOPE-ARGS	The SCOPE (cluster or node) for the enabler to be installed to.	*-cluster SharedListCluser*
	ENABLER_SIBUS _STORE_TYPE	FILESTORE, DATASTORE	*DATASTORE*
	ENABLER_REP_VALUES	xcapRoot, superAdminUser, superAdminPassword	*xcapRoot=http://localhost:9080/services, superAdminUser=superadmin, superAdminPassword=password*
	ENABLER_DB_TYPE	DB2, Oracle	*db2*

Table 9. response file properties (continued)

response section	property	valid values	example value
	ENABLER_DB_HOST	The host name of the database server to be used for the store	*localhost*
	ENABLER_DB_PORT	The port of the database server to be used for the store	*50000*
	ENABLER_DB _DRIVER_PATH	The path to the database client driver JAR files on the WebSphere Application Server host	*/opt/ibm/db2/V9.1/java*
	ENABLER_DB_NAME	The name of the database to be used for the store	*XMDS*
	ENABLER_DB_USER	The user with permissions for the database	*db2inst1*
	ENABLER_DB _PASSWORD	The password of the user with permissions for the database	*db2admin*
Additional Datasources			
	SIBUS_DB_TYPE	DB2, Oracle	*db2*
	SIBUS_DB_HOST	The host name of the database server to be used for the store	*localhost*
	SIBUS_DB_PORT	The port of the database server to be used for the store	*50000*
	SIBUS_DB _DRIVER_PATH	The path to the database client driver JAR files on the WebSphere Application Server host	*/opt/ibm/db2/V9.1/java*
	SIBUS_DB_NAME	The name of the database to be used for the store	*XMDSSIB*
	SIBUS_DB_USER	The user with permissions for the database	*db2inst1*
	SIBUS_DB _PASSWORD	The password of the user with permissions for the database	*db2admin*
	USAGE_RECORD _DB_TYPE	DB2, Oracle	*db2*
	USAGE_RECORD _DB_HOST	The host name of the database server to be used for the store	*localhost*
	USAGE_RECORD _DB_PORT	The port of the database server to be used for the store	*50000*

Table 9. response file properties (continued)

response section	property	valid values	example value
	USAGE_RECORD _DB_DRIVER_PATH	The path to the database client driver JAR files on the WebSphere Application Server host	*/opt/ibm/db2/V9.1/java*
	USAGE_RECORD _DB_NAME	The name of the database to be used for the store	*XMDURDB*
	USAGE_RECORD _DB_USER	The user with permissions for the database	*db2inst1*
	USAGE_RECORD _DB_PASSWORD	The password of the user with permissions for the database	*db2admin*

4. Save the edited document.

Running the silent installation

Installing the IBM XDMS product using silent installation refers to using a response file to supply installation options without user interaction.

Before you begin

You must have met the following requirements:

- WebSphere Application Server Network Deployment, version 7.0.0.1 has been installed with a deployment manager with at least one managed node.
- You have federated all nodes into the deployment manager cell.
- A WebSphere user account repository is created. A federated LDAP repository is recommended. See the WebSphere Application Server Information Center for more instructions.
- IBM DB2 Enterprise Server Edition 9.5 FixPak 1 has been installed along with the license for pureXML.
- The IBM DB2 Enterprise Server Edition 9.5 FixPak 1 client is in the same path on every WebSphere Application Server node in the cluster. This allows the WebSphere variables for the JDBC driver to be declared at the Cell level.
- Oracle Database, version 11.1.0.7 has been installed.
- The Oracle Database, version 11.1.0.7 client is in the same path on every WebSphere Application Server node in the cluster. This allows the WebSphere variables for the JDBC driver to be declared at the Cell level.
- You have created the clusters for each IBM XDMS application you plan to install.
- You have configured the installation by changing the values of the parameters in the response file. Refer to the topic *Customizing the response file* for details.

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.

To perform the silent installation, follow these steps on the deployment manager node.

1. Insert the XDMS installation CD into the CD-ROM drive.
2. Select a umask that would allow the owner to read/write to the files, and allow others to access them according to the prevailing system policy. For root, a umask of 022 is recommended. For non-root users a umask of 002 or 022 could be used, depending on whether or not the users share the group.
3. Change directories (`cd`) to the CD-ROM drive.
4. Set the proper file permissions on the installer by entering the following from the command prompt:

   ```
   chmod 755 setup
   ```
5. Run the silent installation using your customized response file. Issue the following command:

   ```
   ./setup -i silent -f response_file_path/xdms_install.rsp
   ```

   Where:

   response_file_path is the directory path to the customized response file.
6. Start the servers for the installed application.

   Note: During the startup, you can safely ignore any error messages that refer to problems with the UsageRecordWriter.properties file.

Configuring the Aggregation Proxy

If you have installed the Aggregation Proxy, you must perform additional configuration for it.

Configuring the Aggregation Proxy TAI

Configure the Trust Association Interceptor (TAI) (`com.ibm.glm.http.security.tai.HttpDigestTAI`) for Aggregation Proxy so that it will pass proper credentials to IBM XDMS.

Before you begin

Before you proceed, make sure that:

- The file `AggProxyTai.jar` is located in the following directory: *was_root*`/lib/ext`.

  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`
- WebSphere Application Server Network Deployment security is properly configured as either standalone or federated IBM Directory Server on a separate standalone machine.

About this task

Perform the following steps to configure the TAI:

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. (Omit the password if security is not enabled.)

c. Click **Log in**.

2. Click **Security** → **Global security**.

   a. Under Authentication, expand **Web and SIP security** and click **Trust association**.

   b. Click **Enable trust association**.

   c. Under Additional Properties, click **Interceptors**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Security** → **Secure administration, applications, and infrastructure** → **Web Security** → **Trust Association** → **Interceptors**.

3. Click `com.ibm.glm.http.security.tai.HttpDigestTAI`.

4. Click **Custom properties**.

5. Configure the custom properties:

```
Parameter: LdapAuthDn
Example Value: cn=root
Explanation: Pulled from configuration for a Stand-alone LDAP. Most implementations
typically use cn=root but any user with root or superadmin access to all users
defined in the LDAP can be used

Parameter: LdapAuthPw
Example Value: LDAPAUTH_PASSWORD
Explanation: LDAP password of the user specified by the LdapAuthDn parameter

Parameter: LdapBaseDn
Example Value: dc=wasusers
Explanation: Root entry for users matching the value specified by the LdapUserFilter
parameter. In this example it is the root entry for which the entries of object type
inetOrgPerson can be found

Parameter: LdapHost
Example Value: LDAP_HOST
Explanation: Host name of LDAP server

Parameter: LdapPort
Example Value: 389
Explanation: The port on which the LDAP server is listening

Parameter: LdapUserFilter
Example Value: (&(uid=%v)(objectclass=inetOrgPerson))
Explanation: Used to search for a specific user identified by the %v. In this case,
it searches for a user with uid=%v and LDAP user entry type inetOrgperson

Parameter: RetryCount
Example Value: 3
Explanation: Number of times a user is challenged for valid credentials before a
401 Unauthorized response is returned. Note that session affinity must be configured
for the retry count to be remembered for a user.

Parameter: auth.int.enable
Example Value: false
```

```
Explanation: Specifies the auth-int quality of protection (QOP) for digest
authentication. Digest authentication defines two types of QOP: auth and auth-int.
By default False is set to indicate auth QOP. When this custom property is set to
True, the highest level of protection is used, which is the auth-int QOP.

Parameter: XCAPServerContextRoot
Example Value: services
Explanation: Defines the context root for which the digest authentication TAI will
apply.  By default the context root is "services".  If you change the context root
of the Aggregation Proxy, modify this property to match the new context root

Parameter: HttpDigestRealm
Example Value: http.digest.realm
Explnation: Specifies the authentication realm. By default the digest realm is
"http.digest.realm"
```

6. Click **Apply**.
7. Click **Save** to save changes to the master configuration.

Configuring Aggregation Proxy ports

Verify that the virtual host assigned to the Aggregation Proxy application has the correct port definition for the HTTP port on which you want incoming requests to be accepted.

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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. In the navigation pane, click **Applications** → **Enterprise Applications** → **AggregationProxy** → **Virtual hosts**. Note the name of the virtual host that is assigned to the AggProxyWeb module, for example *default_host*.
3. Verify that *default_host* is accessible outside of WebSphere Application Server.
4. 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**.
5. Click **Aggregation Proxy=″AggProxyCluster″** → **Cluster members** → **Application server(s) name** → **Ports.** Note the port number of the *WC_defaulthost* which must be defined in the virtual host.
6. Verify the location of the virtual host you found in step 2 (default_host). Example: *WC_defaulthost 9082*.
7. Navigate to **Virtual Hosts** → ***virtual host=″default_host″*** → **Virtual Hosts** → **Host Aliases** and ensure that the port number you found in step 5 has been added to the list of ports. If not, add it.

Configuring the resource providers for the Aggregation Proxy

If you do not use the default configurations that are provided during installation, you will need to configure the resource provider for the Aggregation Proxy XDMS.

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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. Click **Resources** → **Resource Environment** → **Resource Environment Providers**.
3. Select the **Cluster=AggProxyCluster** scope in the drop-down list.
4. Select **AggProxyREP** from the list of REPs to edit the values.

Example

Example Resource Environment Provider values:

Table 10. Aggregation Proxy properties

Property name	Value	Type	Required?	Description
HTTPS_PROXY_XDMS	false	Boolean	yes	Whether or not this Aggregation Proxy's backend XDMS instances are accessed by an `https` URL
PROXY_ROOT	http://proxy.com:9082	string	yes	The portion of the Aggregation Proxy's URL ending with the context root
THREEGPP_IMS	false	Boolean	yes	Whether or not the 3GPP-GAA is present. If false, `Xdms-Asserted-Identity` is used
XCAP_CACHE_TIMEOUT	180	Integer	yes	Time in seconds that the XCAP-CAPS of each backend XDMS instance should remain in WebSphere's dynamic cache

Table 10. Aggregation Proxy properties (continued)

Property name	Value	Type	Required?	Description
superAdminUser	superadmin	String	yes	ID used by the Aggregation Proxy during initialization and for sending an initial XCAP-CAPS request to each backend XDMS instance
superAdminPassword	password	String	yes	Password used by the Aggregation Proxy during initialization
XDMS_URI	http://hostname1:9080/services #http://hostname2:9080/services	String	yes	XCAP root of XDMS, per domain
XDMS_URI!<domain>	http://sharedlist2.com:9080/services/resource-lists	String	yes	Defines the domain for which the list of XDMS enablers is supported
XDMS_URI_HTTPS	https://sharedlist1.com:9080/services #http://presrules1.com:9081/services	String	no	The XUIs of the backend XDMS clusters serviced by the Aggregation Proxy. List all XUIs here, separated by a # symbol
XDMS_URI_HTTPS !<domain>	https://sharedlist2.com:9080/services/resource-lists	String	no	Defines the secure domain for which the list of XDMS enablers is supported
alarmInterval	15	String	no	Time in minutes between alarm notifications for the same alarm (recommended value: 15)
BAD_XDMS_POLLING_INTERVAL	20	Integer	yes	Time in seconds for polling XDMS instances that have never been online
MAX_PROXY_THREAD_POOL_SIZE	60	Integer	yes	Maximum number of threads in the thread pool used to send proxy requests to XDMSes

Configuring basic routing

Configure just the basic routing when domain partitioning and routing are not required.

About this task

Configuration is performed using 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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. Click **Resources** → **Resource Environment** → **Resource Environment Providers**.
3. Go to the Resource Provider Environment for the Aggregation Proxy, and configure the basic routing.
   a. Click **Custom Properties** under additional properties.
   b. Click **New**. If you are not using HTTPS, then create the following minimum custom properties:

Table 11. Minimum Resource Environment Provider Custom Properties for Basic Routing

Property	Example Value	Description	Notes
PROXY_ROOT	http://example.com/services	Context root of Aggregation Proxy	
HTTPS_PROXY_XDMS	false	Enable/disable HTTPS	
XDMS_URI	http://sharedlist1.com:9080/services# http://presrules1.com:9081/services	The list of the backend XDMS enablers serviced by the Aggregation Proxy.	List all enablers here, separated by a # symbol. Enablers listed here are also known as the ″Default″ route.

If you are using HTTPS, you need to add the following custom properties instead:

Table 12. Minimum Resource Environment Provider Custom Properties for Secure HTTPS Basic Routing

Property	Example Value	Description	Notes
PROXY_ROOT	http://example.com/services	Context root of Aggregation Proxy	
HTTPS_PROXY_XDMS	true		Must be enabled when using HTTPS

Table 12. Minimum Resource Environment Provider Custom Properties for Secure HTTPS Basic Routing (continued)

Property	Example Value	Description	Notes
XDMS_URI_HTTPS	https://sharedlist1.com:9080/services#http://presrules1.com:9081/services	The XUIs of the backend XDMS clusters serviced by the Aggregation Proxy.	List all XUIs here, separated by a # symbol.

Configuring advanced routing

Configure advanced routing when domain partitioning and matching are required.

About this task

Configuration is performed using 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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. Click **Resources** → **Resource Environment** → **Resource Environment Providers**.
3. Navigate to the Resource Provider Environment for Aggregation Proxy and configure the basic routing.
   a. Click **Custom Properties** under additional properties.
   b. Click **New**. If you are not using HTTPS, then create the following minimum custom properties:

Table 13. Minimum Resource Environment Provider Custom Properties for Advanced Routing

Property	Example Value	Description	Notes
PROXY_ROOT	http://example.com/services	Context root of Aggregation Proxy	
HTTPS_PROXY_XDMS	false	Enable/disable HTTPS	
XDMS_URI	http://sharedlist1.com:9080/services#http://presrules1.com:9081/services	The list of the backend XDMS enablers serviced by the Aggregation Proxy.	List all enablers here, separated by a # symbol. Enablers listed here are also known as the "Default" route.

Table 13. Minimum Resource Environment Provider Custom Properties for Advanced Routing (continued)

Property	Example Value	Description	Notes
XDMS_URI!<domain>	http://sharedlist2.com:9080/services/resource-lists	Defines the domain for which the list of XDMS enablers is supported	Add one property per domain you are serving. Example Property name : 4. XDMS_URI!example.com

If you are using HTTPS, you need to add the following custom properties instead:

Table 14. Minimum Resource Environment Provider Custom Properties for Secure HTTPS Advanced Routing

Property	Example Value	Description	Notes
PROXY_ROOT	http://example.com/services	Context root of Aggregation Proxy	
HTTPS_PROXY_XDMS	true		Must be enabled when using HTTPS
XDMS_URI_HTTPS	https://sharedlist1.com:9080/services#http://presrules1.com:9081/services	The XUIs of the backend XDMS clusters serviced by the Aggregation Proxy.	List all XUIs here, separated by a # symbol.
XDMS_URI_HTTPS!<domain>	https://sharedlist2.com:9080/services/resource-lists	Defines the secure domain for which the list of XDMS enablers is supported	

Adding a server to an existing IBM XDMS node

If you have a running system with a IBM XDMS 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

The IBM XDMS installer has been run on the new server node, and IBM XDMS Core files have been installed.

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. (Omit the password if security is not enabled.)

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. Update the SIBus to include a new member
   a. On the deployment manager server, change directories to *was_root*/bin.
   b. Run the following command: `./wsadmin.sh –lang jython –f` *was_root*`/installableApps/xdms/scripts/install/xdmsMessaging.py -create -cluster` *cluster_name* `-busDataSourceJndi` *jndi_name*. For example:

      ```
      ./wsadmin.sh –lang jython
      –f /opt/IBM/Websphere/AppServer/installableApps/xdms/scripts/install/xdmsMessaging.py
      -create –cluster SharedListCluster -busDataSourceJndi jdbc/xdmssib.
      ```
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 *SharedList0*.
   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 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. For Oracle setup only, adjust the JVM settings by following the steps in the topic "Configuring IBM XDMS to work with Oracle Database" on page 49.
7. Stop and restart the application server, nodes, and deployment manager.

Uninstalling IBM XDMS

If you no longer need the IBM WebSphere XML Document Management Server Component, follow these procedures to uninstall it.

Before you begin

Before uninstalling the IBM XDMS each of the managed nodes where IBM XDMS has been installed must, be started at least once prior to uninstall.

Uninstalling IBM XDMS using the interactive uninstaller

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

About this task

Use the following procedure to uninstall the IBM XDMS program code, including the Shared List and Presence Rules enterprise applications.

1. The uninstaller is created in the *was_root*/Uninstall_XDMS directory during the installation. Change directories (cd) to this directory.
2. Issue the following command to launch the interactive uninstall wizard:
   ```
   ./uninstall
   ```
3. Select the WebSphere Application Server instance and profile.
4. Follow the wizard to select the subcomponents that you want to uninstall, and follow the wizard prompts.
5. Optional: To examine the logs, view the file *was_root*/logs/XDMS/xdmsUninstall.log log.

What to do next

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

Uninstalling IBM XDMS silently

As an alternative to using the wizard, the IBM WebSphere XML Document Management 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 IBM XDMS 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, `xdms_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.

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/xdms/samples`.

   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, `xdms_uninstall.rsp`, in a text editor.
3. Edit the response file, modifying the following values as needed. The following table lists the response file parameters that are used to uninstall.

Table 15. Response file properties

response section	property	valid values	example value
License			
	LICENSE_ACCEPTED=	Must equal true to use installer	*true*
WebSphere Variables			
	WAS_HOME=	valid path on the local machine with no trailing slash	*/opt/IBM/WebSphere/AppServer*
	WAS_HOST=	Valid hostname or IP address	*localhost*
	WAS_PORT=	Valid port number	*8879*
	WAS_SECURED=	0 for false, 1 for true	*1*
	WAS_USER=	Valid WebSphere Application Server administrative user name	*was_admin*
	WAS_PASSWORD=	Valid WebSphere Application Server administrative user's password	*waspass*

Table 15. Response file properties (continued)

response section	property	valid values	example value
	REMOTE_NODE	Is this maintenance being performed on a standalone machine (0), deployment manager (0), or managed node (1)	*0* or *1*
	PROFILE_HOME	If this maintenance is being performed on the deployment manager machine, or a machine with a standalone node, then you must specify the Dmgr or node's profile home directory	*/opt/IBM/WebSphere/AppServer/profiles/Dmgr01*
Installer Actions			
	CORE_FILES_ACTION=	Action to take for IBM XDMS program code: INSTALL, UNINSTALL, UPGRADE, or NONE	*UNINSTALL*
	AGGPROXY_ACTION=	Action to take for the Aggregation Proxy: INSTALL, UNINSTALL, UPGRADE, or NONE	*UNINSTALL*
	SHAREDLISTXDMS_ACTION=	Action to take for the Shared List XDMS application: INSTALL, UNINSTALL, UPGRADE, or NONE	*UNINSTALL*
	PRESENCERULESXDMS_ACTION=	Action to take for the Presence Rules XDMS application: INSTALL, UNINSTALL, UPGRADE, or NONE	*UNINSTALL*

4. Save the edited file.
5. Copy the edited file to each cluster member and deployment manager where you plan to uninstall the IBM XDMS 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 IBM XDMS product silently, without user interaction.

Before you begin

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

About this task

Follow these steps to uninstall IBM XDMS in silent mode:

1. The uninstaller is created in the *was_root*/Uninstall_XDMS directory during the installation. Change directories (cd) to this directory.
2. To uninstall the IBM XDMS product using your custom response file, issue the following command. For example:

   ```
   ./uninstall -i silent -f response_file_path/xdms_uninstall.rsp
   ```

   where *response_file_path* is the directory path to your response file.
3. Optional: To examine the logs, view the file *was_root*`/logs/XDMS/xdmsUninstall.log log`.

What to do next

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

Removing the Resource Environment Providers

After uninstalling the IBM XDMS program code and the XDMS enterprise applications, remove Resource Environment Providers (REPs) for the Aggregation Proxy and for the enterprise applications.

Before you begin

Uninstall all XDMS enterprise applications and the Aggregation Proxy.

About this task

To remove the Resource Environment Providers, complete the following steps at the deployment manager node:

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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
2. Remove the Resource Environment Provider for the Aggregation Proxy:

a. Navigate to **Resources** → **Resource Environment** → **Resource Environment Provider**.
b. In the Scope drop down list, select the Aggregation Proxy cluster. Example: AggProxyCluster.
c. Select **AggProxyREP** and click **Delete**.

3. Remove the Resource Environment Providers for all XDMS enterprise applications:
   a. Return to **Resources** → **Resource Environment** → **Resource Environment Provider**.
   b. In the Scope drop-down list, select the cluster. Example: SharedListCluster.
   c. Select all of the REPs, and click **Delete**.
4. Save your changes to the master configuration.
   a. Click **System administration** → **Save Changes to Master Repository**.
   b. Select **Synchronize changes with Nodes** and click **Save**.
5. Stop the node agents and deployment manager.
   a. Stop all node agents.
   b. Stop the deployment manager.

Uninstalling the Trust Association Interceptor Security Component

The Trust Association Interceptor feature is uninstalled from the WebSphere Application Server Administration Console.

About this task

Perform the following steps to uninstall the interceptor:

1. Delete the file `DHAIMSConnectorTAI.jar` from the *`was_root`*`/lib/ext` directory.

   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. 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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
3. 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**.

4. In the Global security window, under Authentication, click **Web and SIP security** → **Trust association**.

   Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Web security** → **Trust association**.

5. Optional: In the Trust Association window, disable trust association:
   a. Clear the **Enable trust association** check box.
   b. Click **OK**.
   c. Click **Save**.
6. Return to the Trust Association window.
7. Under Additional Properties, click **Interceptors**.
8. Select **com.ibm.imsconnector.tai.HttpInterceptor** and **com.ibm.imsconnector.tai.SipInterceptor**.
9. Click **Delete**.
10. Click **Save**.
11. Restart the server.

Uninstalling the Aggregation Proxy interceptor security component

The Aggregation Proxy interceptor feature is uninstalled from the WebSphere Application Server Administration Console.

About this task

Perform the following steps to uninstall the interceptor:

1. Delete the `AggProxyTai.jar` from the *`was_root`*`/lib/ext`.

   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. 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. (Omit the password if security is not enabled.)
   c. Click **Log in**.
3. Click **Security** → **Global security**.

a. Under Authentication, expand **Web and SIP security** and click **Trust association**.
b. Deselect **Enable trust association**.

Note: If you are using WebSphere Application Server version 6.1.0.x, reach this window by clicking **Security** → **Secure administration, applications, and infrastructure** → **Web Security** → **Trust Association**.

Note: You can keep the trust association enabled and then uninstall the Aggregation Proxy interceptor by using the following instructions.

4. Click **OK**.
5. Click **Save**.
6. Return to the Trust Association window.
7. Under Additional Properties, click **Intercepters**.
8. Select **com.ibm.glm.http.security.tai.HttpDigestTAI**.
9. Click **Delete**.
10. Click **Save**.
11. Restart the server.

Dropping the databases

After you have uninstalled the IBM XDMS program code and other components, you can drop the databases.

Dropping the DB2 databases

Follow these instructions to drop the DB2 databases that were used by IBM XDMS.

About this task

The databases include one or more of the following:

- Shared List database
- Shared Group database
- Shared Profile database
- Shared Policy database
- Shared Enablers database
- Presence Rules database
- JMS SIBus database
- Usage Record database

1. Log on to the database server as the instance owner.
2. For each database, run the following command, where *db_name* is the name of the database:

```
db2 drop database db_name
```

Dropping the Oracle databases

Follow these instructions to drop the Oracle databases that were used by IBM XDMS.

About this task

The databases include one or more of the following:

- Shared List database

- Shared Group database
- Shared Profile database
- Shared Policy database
- Shared Enablers database
- Presence Rules database
- JMS SIBus database
- Usage Record database

1. Log on to the database server as the database administrator.
2. Connect to the first database by running the following command:

   ```
   sqlplus $dbuser/$dbuserPW@$db_name
   ```

   where:

   dbsser is the user ID for the authorized user.

   dbuserPW is the password for the authorized user.

   db_name is the name of the usage records database, for example XDMSUR.
3. Remove the database tables by running the DROP TABLE command for each table that is defined.
4. Drop the database using the standard Oracle procedure.
5. Repeat steps 2 through 4 for each of the other databases.

Chapter 4. Administering IBM XDMS

From the command-line interface, you can perform administrative tasks for IBM XDMS such as adding, deleting, and modifying policy documents.

System Management

Administrators perform a variety of system management tasks, using the WebSphere console.

About this task

You may need to perform the following administrative tasks:

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. (Omit the password if security is not enabled.)

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. (Omit the password if security is not enabled.)
   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. (Omit the password if security is not enabled.)
   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. (Omit the password if security is not enabled.)

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. (Omit the password if security is not enabled.)
   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. (Omit the password if security is not enabled.)
   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*.

IBM XDMS user identity management

User identity management and access levels for IBM XDMS should be handled using Policy documents.

About this task

The following information describes how to use policy-based authorization with IBM XDMS:

Policy based authorization

Policy based authorization allows users to specify authorization rules by sending XCAP requests to create, update, or delete authorization documents.

The authorization policy documents are based on the IETF common policy specification draft-ietf-geopriv-common-policy-11 and the OMA common policy extensions documented in section 5.2.2 of the XDM_Core_V2 specification.

The common policy documents contain a set of rules. Each rule contains three elements: *conditions*, *transformations* (not used in IBM XDMS), and *actions* (also not supported by IBM XDMS).

The IETF geopriv common policy provides ways to specify the following identities:

- *one*: One specific authenticated identity. An identity element can specify multiple *one* elements.
- *many*: Matches all authenticated identities including the anonymous identity. Can include *except* elements to exclude specific identities.

The OMA XDM Core extends the *conditions* element to allow references besides the *identity* element:

- *external-list*: A reference to a resource-list that contains multiple authenticated identities.
- *anonymous-request*: A reference to an anonymous identity. This may or may not refer to unauthenticated users. It is determined by the Trust Association Interceptor (TAI) to create the special **anonymous.invalid** principal when receiving a request from a requester.

The IBM XDMS provides a way to specify authorization policies at various levels of the XCAP URI path hierarchy. The URI and name of the document will determine the access type that it grants. The AUID for authorization policy documents will depend on the AUID of the document being protected.

Here are the various levels where authorization policies can be applied:

Table 16. Access levels for authorization policies

Access type	**Description**	**Policy Document XCAP URI**
Global directory	Who has access to all documents in the global directory.	http://*xcapHost*:*xcapPort*/services/com.ibm.*auid*-acls/global/directory.xml
Global file	Who has access to a specific document in the global directory.	http://*xcapHost*:*xcapPort*/services/com.ibm.*auid*-acls/global/<global_document>
XUI domain	Who has access to all documents that match an XUI domain/subdomain. The XUI for the domain is specified with the domain scheme such as "domain:<subdomain>." This prevents name collisions with a real user within the XUI.	http://*xcapHost*:*xcapPort*/services/com.ibm.*auid*-acls/users/domain:<subdomain>/directory.xml
XUI directory	Who has access to all documents in the XUI directory.	http://*xcapHost*:*xcapPort*/services/com.ibm.*auid*-acls/users/*XUI*/directory.xml
XUI file	Who has access to a specific document in the XUI directory.	http://*xcapHost*:*xcapPort*/services/com.ibm.*auid*-acsl/users/*XUI*/*XUI_document*

The AUID for authorization policy documents will depend on the AUID of the document being protected. The suggested naming convention is com.ibm.*auid*-acls. The name of the authorization policy AUID for the resource-lists AUID is com.ibm.resource-lists-acls.

The IBM authorization policy includes a schema that describes the special authorization *rules* that are allowed for an authorization policy. The *rule* attribute *id* describes the access levels and can contain the following attribute values:

- read: Grants the read permission to read the contents of the target document or element. Also includes the ability to subscribe to the document.
- write: Grants the write permission to write to the contents of the target document or element.
- delete: Grants the delete permission to delete the contents of the target document or element.

- admin: Grants the admin permission which encompasses the read, write, and delete permissions and also allows the ability to modify the authorization policy document.

The following elements are listed in the Open Mobile Alliance specifications for policy documents but are not supported by the IBM XDMS:

- *sphere*: No context-specific rules allowed for authorization rules
- *validity*: No time-based conditions allowed for authorization rules
- *actions*: No special actions needed because permissions are defined in the rule ID
- *transformations*: No special transformations allowed for authorization rules
- *media*: No special media characteristics affect authorization rules

You can create policy documents using any editor you choose.

Adding new policy documents

Add new policy documents using the xcap_put.sh request.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you have JDK1.6.0 SR 3 installed and configured in your system path variables.

About this task

Create or edit an XDM policy document and post it to the XDMS.

1. Reach the command client by typing the following from the command-line directory: `cd was_root/installableApps/xdms/client/`

   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. Type the XCAP Put request.

   ```
   ./xcap_put.sh -user user_id -password password -filename file_name -content_type application/auth
   ```

   For example, to post a Resource List policy document name `resourcelistspolicy.xml` to the XDMS server for example.com, type the following parameters on a single line.

   ```
   ./xcap_put.sh -user admin -password adminpassword -filename path/resourcelistspolicy.xml -content
   ```

3. Press Enter to send the request. The new policy document is posted to the XDMS in the specified location – in this case, the `global` directory to protect the document named `resourcelistspolicy.xml`.

What to do next

Policy documents can be retrieved or deleted using the XCAP Get and XCAP Delete commands.

Granting a single identity admin access to a specific document

You can grant a single identity with administrative permission access to a specific document in their home directory.

Before you begin

Preconditions:

1. User John Doe (sip:john.doe@us.example.com) owns a buddy list document in his home directory with the following XCAP URI:

   http://xdms.example.com:9080/services/resource-lists/users/sip:john.doe@us.example.com/buddylist.xml

2. John has a friend Bob (sip:bob.cool@us.example.com) who would like admin access to John's buddy list.

About this task

In order for John to provide Bob access, John must create an authorization policy document that complies with the IETF common policy specification. The authorization policy document must look like this to provide administrative access to Bob.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
 <rule id="admin">
  <conditions>
   <identity>
    <one id="sip:bob.cool@us.example.com" />
   </identity>
  </conditions>
 </rule>
</ruleset>
```

The authorization policy document must be created with the exact XCAP URI as the original resource-lists document, except that the AUID is substituted with the access control list (ACL) AUID: com.ibm.resource-lists-acls.

```
http://xdms.example.com:9080/services/com.ibm.resource-lists-acls/users/sip:john.doe@us.example.co
```

This defines the authorization policy document that grants administrative access to Bob for the corresponding buddy list document that is stored in the resource-lists AUID.

Create or edit an XDM policy document and post it to the XDMS.

1. Reach the command client by typing the following from the command line: `cd` *`was_root`*`/installableApps/xdms/client/`

   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. Type the XCAP Put request.

   ```
   ./xcap_put.sh -user user_id -password password -filename file_name -content_type application/a
   ```

   For example, to put the preceding example to the XDMS server for xdms.example.com, type the following parameters on a single line:

   ```
   ./xcap_put.sh -user sip:john.doe@us.example.com -password password -filename samples/AuthPolic
   ```

3. Press Enter to send the request. The new policy document is put to the XDMS in the specified location – in this case, for the XCAP URI that corresponds to the authorization policy document that protects John's buddy list document.

What to do next

Policy documents can be retrieved or deleted using the XCAP Get and XCAP Delete commands.

Granting many identities read access to all documents in a directory

Users can grant multiple identities read access to all documents in their home directory.

Before you begin

Preconditions:

1. John would like to share all documents in his directory except with hacker Sam (sip:sam.hacker@us.example.com).

In order for John to provide everyone read access except for Sam, John must create an authorization policy document that complies with the IETF common policy specification. The authorization policy document must look like this to provide read access for everyone except Sam.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
 <rule id="read">
  <conditions>
   <identity>
    <many>
     <except id="sip:sam.hacker@us.example.com"/>
    </many>
</identity>
  </conditions>
 </rule>
</ruleset>
```

The authorization policy document must be created with the XCAP URI to a special `directory.xml` in John's home directory with the access control list (ACL) AUID:

```
com.ibm.resource-lists-acls.http://xdms.example.com:9080/services/com.ibm.resource-lists-acls/users/
```

This defines the authorization policy document that grants read access to everyone (many) except for Sam to read any document stored in John's home directory for the corresponding resource-lists AUID.

Before using the XDMS client and XCAP requests, make sure that you have JDK version 1.6.0 SR 3 installed and configured in your system path variables.

About this task

Create or edit an XDM policy document and post it to the XDMS.

1. Reach the command client by typing the following from the command line:`cd` *`was_root`*`/installableApps/xdms/client/`

   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. Type the XCAP Put request.

   ```
   ./xcap_put.sh -user user_id -password password -filename file_name -content_type application/a
   ```

   For example, to put the preceding example to the XDMS server for xdms.example.com, type the following parameters on a single line:

   ```
   /xcap_put.sh -user sip:john.doe@us.example.com -password password -filename samples/AuthPolicy
   ```

3. Press Enter to send the request. The new policy document is put to the XDMS in the specified location – in this case, for the XCAP URI that corresponds to the authorization policy document that protects all documents in John's home directory.

What to do next

Policy documents can be retrieved or deleted using the XCAP Get and XCAP Delete commands.

Granting a group of identities write access to all documents in a domain

You can grant a group of identities write access to all documents in a domain.

Before you begin

Preconditions:

1. User SuperAdmin (superadmin) owns a domain user list document in his home directory with the following XCAP URI: `http://xdms.example.com:9080/services/resource-lists/users/superadmin/domainUserList.xml`
2. The domain user list document contains a list named "ServicesForUSDomain". This special list contains a list of service IDs that SuperAdmin will provision to grant write access to all documents within a certain user domain.

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
 <list name="ServicesForUSDomain">
  <entry uri="sip:service1@us.example.com"/>
  <entry uri="sip:service2@us.example.com"/>
 </list>
</resource-lists>
```

About this task

In order for SuperAdmin to provide write access to service IDs listed in the domain user list document, SuperAdmin must create an authorization policy document that complies with the IETF common policy specification. The authorization policy document must be similar to the following example, which references the external list *ServicesForUSDomain* that is defined in the domain user list document.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy"
 xmlns:oma="urn:oma:xml:xdm:common-policy">
 <rule id="write">
  <conditions>
   <oma:external-list>
    <oma:entry
     anc="http://xdms.example.com:9080/services/resource-lists/superadmin/domainUserList.xml/~~/re
   </oma:external-list>
  </conditions>
```

```
  <actions />
  <transformations />
 </rule>
</ruleset>
```

Note:

Take note of the following:

- The <*oma:external-list*> element contains an `oma` prefix because the element is defined in the OMA schema for common-policy.
- The <*oma:entry*> element contains an anc attribute which is an anchor to a fully qualified resource-lists element. The Node Selector references the specific *ServicesForUSDomain* list.
- The anc attribute must be percent encoded therefore the latter portion of the Node Selector list[@name="ServicesForUSDomain"] is percent encoded into list%5b@name=%22ServicesForUSDomain%22%5d.

The authorization policy document must be created with the XCAP URI to a special directory.xml within a special users XUI named domain:us.example.com with the access control list (ACL) AUID which is com.ibm.resource-lists-acls.`http://xdms.example.com:9080/services/com.ibm.resource-lists-acls/users/domain:us.example.com/directory.xml`

This defines the authorization policy document that grants write access to all service IDs ,within the domain user list document, to write to any document stored in any users directory that is under the domain us.example.com (including sub-domains) for the corresponding resource-lists AUID.

Before using the XDMS client and XCAP requests, make sure that you have JDK1.5.0 SR 5 installed and configured in your system path variables.

Create or edit an XDM policy document and post it to the XDMS.

1. Reach the command client by typing the following from the command line: `cd` *`was_root`*`/installableApps/xdms/client/`

   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. Type the XCAP Put request:

   ```
   ./xcap_put.sh -user user_id -password password -filename file_name -content_type application/auth
   ```

   For example, to put the file to the XDMS server for xdms.example.com, type the following parameters on a single line.

   ```
   ./xcap_put.sh -user superadmin -password password -filename samples/AuthPolicyExternalListWrite.x
   ```

3. Press Enter to send the request. The new policy document is posted to the XDMS in the specified location–in this case, for the XCAP URI that corresponds to the special `directory.xml` authorization policy document that protects the us.example.com domain.

What to do next

Policy documents can be retrieved or deleted using the XCAP Get and XCAP Delete commands.

Granting node level authorization

Authorize user access to specific nodes within documents.

About this task

IBM XDMSusers can specify permissions at the node level within a XML document, using the rule ids: read, write, delete and admin. In order to specify one of those rules for a single node, the XPATH to that URI can be appended to the end of the rule id. This will ensure that the permission in the rule id, and its conditions, are only applied to the node of the document specified, and not to the entire document.

The following document would give user *sip:xdms2@us.acme.com* read access to the list **coworkers** and write access to the list **friends** in any document this ACL document authorizes.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
 <rule id="read/resource-lists/list[@name=%22coworkers%22]">
  <conditions>
   <identity>
    <one id="sip:xdms2@us.acme.com" />
   </identity>
  </conditions>
  <actions />
  <transformations />
 </rule>
 <rule id="write/resource-lists/list[@name=%22friends%22]">
  <conditions>
   <identity>
    <one id="sip:xdms2@us.acme.com" />
   </identity>
  </conditions>
  <actions />
  <transformations />
 </rule>
</ruleset>
```

Managing documents

You can view, add, edit, and delete documents from the XDMS using XCAP requests.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you have JDK1.6.0 SR 3 from IBM, installed and configured in your system path variables.

Note: You must be authorized to access the XCAP servlet or be an authenticated WebSphere Application Server user, depending on the selection made for the Security role mapping to the user/group option during the installation process. If security is enabled, IBM WebSphere XDMS Component will authenticate your user ID and password. If you do not specify the password, and security is enabled, then XDMS will prompt you and hide the text during input. If security is not enabled, you may still enter a user ID and password, but IBM XDMS will not authenticate them.

About this task

Follow the instructions in the following sections to manage documents.

Using IBM XDMS to store and manage documents

IBM WebSphere XML Document Management Server Component enables storage, access, and manipulation of XML documents stored in a central repository.

IBM XDMS stores documents in central DB2 databases. The XDMS databases contain tables identified and named by an Application Unique Identifier (AUID). The stored documents can then be accessed and manipulated using one of two protocols:

- XML Configuration Access Protocol (XCAP) over HTTP
- Session Initiation Protocol (SIP)

IBM XDMS reduces overhead and bandwidth by enabling the selection and manipulation of specific elements within an XML document instead of the entire document.

The XCAP specification allows the specification of a node selector, which is a subset of XPath.

Using XCAP to store and manage documents

The XCAP specification defines how an HTTP Web address can identify the way XML documents are stored on an XCAP server. It also defines how the URI can be used to identify entire XML documents, individual elements, or XML attributes that can be retrieved, updated, or deleted.

The XML documents are stored on an XCAP server, which acts as a repository for documents with each application having access to multiple documents for each user. Each XCAP resource on an XCAP server has an associated application. For the associated application to use the XCAP resources, the application must have the following usage information:

- An Application Unique ID (AUID), which uniquely identifies the application usage, must be created.
- An XML schema must be defined.
- The default namespace binding, which maps the namespace prefixes to the namespace URIs, must be set.
- The MIME type of the document must be defined.
- The XCAP server must be able to validate the content of each XCAP document that is being modified.
- The XML documents to be managed for an application must be well-formed.
- Naming conventions for XCAP client URIs must be set.
- Resource interdependencies, how changes to one resource will affect other resources, have to be determined.

XCAP URI

The XCAP URI identifies the type of XML that is being stored or accessed, a unique identification of the XML document, and optionally, an XPath expression that can identify a specific XML node that is to be stored, deleted, or retrieved.

The general form of an XCAP URI used to access an XML document is as follows:

XCAP Root/AUID/Document Selector/~~/Node Selector

- *XCAP Root* identifies the HTTP request host, port, and context root.

  - Example: `http://myhost.ibm.com:9080/services/`
- *AUID* is the XCAP Application Unique ID. This identifies the type of XML document. Some AUIDs are defined within XCAP Usage specifications, user defined AUIDs can be developed using the XDMS toolkit.
  - Examples: resource-lists or rls-services
- *Document Selector* is the portion of the URI which identifies the specific document to be stored or accessed. XCAP documents are segregated into two tree branches. The global branch contains documents which can only be created by administrators, but accessed by anybody. The users branch contains user-specific documents.
- `/~~/` is a separator between the document selector, and an optional Node Selector.
- *Node Selector* is a limited XPath expression that can be used to identify a specific XML element or attribute which is to be updated, retrieved, or deleted.

A complete XCAP URI might look like this:

```
http://myhost.ibm.com:9080/services/rls-services/users/
sip:user1@example.com/exampdoc.xml/~~/exampelement
```

Using the Node Selector

Node Selector is a subset of the XPath expression, used to identify a specific element or attribute of the XML document.

Node Selector enables the selection of XML document elements from the document, element, or attribute level. Using Node Selector you can specify the desired element you want to retrieve, store, or alter.

Example

A document is stored in the XDMS with the following XCAP URI:
`http://xdms.example.com/services/resource-lists/global/example.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="friends">
    <entry uri="sip:friend1@example.com">
      <display-name>My Friend1</display-name>
    </entry>
    <entry uri="sip:friend2@example.com">
      <display-name>My Friend2</display-name>
    </entry>
    <entry uri="sip:friend3@example.com">
      <display-name>My Friend3</display-name>
    </entry>
    <entry uri="sip:friend4@example.com">
      <display-name>My Friend4</display-name>
    </entry>
    <entry uri="sip:friend5@example.com">
      <display-name>My Friend5</display-name>
    </entry>
    <entry uri="sip:friend6@example.com">
      <display-name>My Friend6</display-name>
    </entry>
    <entry uri="sip:friend7@example.com">
      <display-name>My Friend7</display-name>
    </entry>
    <entry uri="sip:friend8@example.com">
      <display-name>My Friend8</display-name>
```

```
    </entry>
    <entry uri="sip:friend9@example.com">
      <display-name>My Friend9</display-name>
    </entry>
  </list>
</resource-lists>
```

In bold, the example shows the document root *resource-lists* with the name space *urn:ietf:params:xml:ns:resource-lists*. Following the document root statement is an element called *list*, and the attribute **name** with the value *friends*. Nested elements named `entry` with the attribute **uri**, in the bolded instance the uri attribute has the value *sip:friend1@example.com*.

Node selector example

A full node selector statement will have the following format:
`XCAP_URL/~~/root-element/element[@name="attributevalue"]`

From the example document in the preceding section, you can construct a node selector statement to select uri=sip:friend1@example.com from the resource-list `friends`:

```
/resource-lists/list[@name="friends"]/entry[@uri="sip:friend1@example.com"]
```

Adding and editing documents

Use the xcap_put.sh request to add or edit documents from the command line interface.

Before you begin

Before using the XCAP client to submit XCAP request to IBM XDMS, make sure that you have JDK1.6.0 SR 3 from IBM, installed and configured in your system path variables.

About this task

Create or edit an XML document that is of a type managed by one of the AUIDs on your IBM XDMS, or use of the example documents located in the *was_root*`/installableApps/xdms/client/` directory.

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. Reach the command client by typing the following from the command line:`cd` *was_root*`/installableApps/xdms/client/`
2. To use the XCAP client to submit a Put request, issue the following command:

   ```
   ./xcap_put.sh -user user_id -password password -filename file_name -content_type mime-type XCAP_U
   ```

   or example, to put the sample TestResourceList.xml to the XDMS server for the user testuser@example.com, type the following parameters on a single line:

   ```
   ./xcap_put.sh -user admin -password adminpassword -filename samples/TestResourceList.xml -content
   ```
3. Run the command to send the request. If the request was to put a new document, it now appears in the specified directory. If the request used a Node Selector to send an update to a document, the new content now appears in the specified document.

Adding and editing elements

Using a node selector statement in the XCAP URI, you can select elements to be added or edited in an existing document.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you have JDK1.6.0 SR 3 installed and configured in your system path variables.

About this task

Create or edit an XDM document of your choice, or use the TestListElement.xml document located in the *was_root*/installableApps/xdms/clients/samples directory and add it to the system using the xcap_put.sh command.

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. Create an xml document containing a new element and save it. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<list name="coworkers" xmlns="urn:ietf:params:xml:ns:resource-lists" xmlns:oau="urn:oma:xml:xd
        <display-name>Co-Workers</display-name>
        <entry uri="sip:employee1@example.com">
                <display-name>Employee #1</display-name>
        </entry>
        <entry uri="sip:employee2@example.com">
                <display-name>Employee #2</display-name>
        </entry>
</list>
```

2. From the command line type: cd *was_root*/installableApps/xdms/client/ to reach the command client.
3. Type the XCAP PUT request.

```
./xcap_put.sh -user user_id -password password -filename file_name -content_type application/x
```

   Note: This is one example of a Node Selector. Refer to IETF RFC 4825 for information about other ways to specify a Node Selector for an element.

   For example:

   To post the sample TestListElement.xml into the TestResourceList.xml document on the XDMS server for xdms1@us.example.com, type the following parameters on a single line:

```
./xcap_put.sh -filename samples/TestListElement.xml -user sip:xdms1@us.acme.com -password xdms
```

4. Run the command to send the request. The new content will now appear in the specified document.

Adding and editing attributes

Using a node selector statement in the XCAP URI you can select attributes to be added or edited in an existing document.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you have JDK1.6.0 SR 3 installed and configured in your system path variables.

About this task

Create or edit an XDM document of your choice, or use the TestAttributeExample.xml document located in the *was_root*/installableApps/xdms/clients/samples directory and add it to the system using the xcap_put.sh command.

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. Modify the TestLangAttrValue.txt document and change the lang attribute value to *en* or any other language value and save it.
2. Switch to the *was_root* directory.
3. Reach the command client by typing the following from the command line:cd *was_root*/installableApps/xdms/client/
4. Type the XCAP Put request.

   ```
   ./xcap_put.sh -user user_id -password password -filename file_name -content_type application/xcap
   ```

   Note: This is one example of a Node Selector. Refer to IETF RFC 4825 for information about other ways to specify a Node Selector for an attribute.

   For example, to post the sample TestLangAttrValue.txt to the XDMS server for example.com, type the following parameters on a single line.

   ```
   ./xcap_put.sh -user sip:xdms1@us.example.com -password password -filename samples/TestLangAttrVal
   ```
5. Press Enter to send the request. The new content now appears in the specified document.

Retrieving documents

Use the xcap_get.sh request to retrieve stored documents.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you have JDK1.6.0 SR 3 installed and configured in your system path variables.

About this task

Determine the name and location of an XDM document you would like to retrieve.

1. Reach the command client by typing the following from the command line:cd *was_root*/installableApps/xdms/client/

   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. Type the XCAP Get request.

   ```
   ./xcap_get.sh -user user_id -password password -output file_name XCAP_URL
   ```

   For example, to retrieve the sample TestResourceList.xml from the XDMS server for example.com, type the following parameters on a single line.

   ```
   ./xcap_get.sh -user admin -password adminpassword -output samples/TestResourceList.xml http://exa
   ```

3. Press Enter to send the request. The document specified in the XCAP_URL for retrieval is stored locally as <file_name>. In the example provided, the file is stored locally as `samples/TestResourceList.xml`.

Deleting documents

Use the xcap_delete.sh request to delete XDMS documents.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you have JDK1.6.0 SR 3 installed and configured in your system path variables.

About this task

Determine the name and location of an XDM document you would like to retrieve.

1. Reach the command client by typing the following from the command line:`cd *was_root*/installableApps/xdms/client/`

   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. Type the XCAP Delete request.

   `./xcap_delete.sh -user *user_id* -password *password* *XCAP_URL*`

   For example, to delete the sample TestResourceList.xml from the XDMS server for example.com, type the following parameters on a single line.

   ```
   ./xcap_delete.sh -user admin -password adminpassword http://example.com/services/resource-list
   ```

3. Press Enter to send the request. The document specified in the XCAP_URL is deleted. In the provided example provided, the document TestResourceList.xml is deleted.

Searching documents

Use the xcap_post.sh request to search documents from the command line interface.

Before you begin

Before using the XDMS client and XCAP requests, make sure that you:

- Have IBM JDK1.5.0 SR 5 installed and configured in your system path variables.

About this task

As defined in the OMA XDM v2.0 specification, IBM XDMS supports the ability to search within stored documents using XQuery. To perform an XQuery search, the client sends a search document to an OMA XDM v2.0 capable search proxy or directly to IBM XDMS by means of an HTTP POST.

Before performing the search, IBM XDMS extracts the AUID from the search document. If the AUID is supported, it verifies if the XQuery in the search document matches XQuery templates defined by the application usage. If the XQuery matches a defined template, the search is then performed.

It is important to note that URIs used for XCAP and XQuery differ. XQuery URIs have the following form:

XCAP Root/org.openmobilealliance.search

XCAP Root: identifies the HTTP request host, port, and context root.

```
http://myhost.ibm.com:9080/services/
```

Create or edit an XDM document of your choice, or use one of the documents located in the *was_root*/installableApps/xdms/client/ directory.

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. Reach the command client by typing the following from the command line:cd *was_root*/installableApps/xdms/client/
2. Type the XCAP Post request.

   ```
   ./xcap_post.sh -user user_id -password password -filename file_name -content_type mime-type XCAP_
   ```

   For example, to post the sample TestSearchResourceList.xml to the XDMS server for example.com, type the following parameters on a single line.

   Note: In order for this sample to work, insert the TestResourceList.xml file by following the steps in the topic *Adding and editing documents*.

   ```
   ./xcap_post.sh -user admin -password adminpassword -filename samples/TestSearchResourceList.xml -
   ```

3. Press Enter to send the request. The TestSearchResourceList.xml search file is sent as an HTTP Post to IBM XDMS. The search results are then returned to the XDMS client.

Managing Document Subscriptions

Use SIP subscription requests for users to want to subscribe to change notifications on XDMS documents.

About this task

Follow the instructions in these topics:

Subscribe and notify

The XCAP component provides a SIP SUBSCRIBE and NOTIFY interface into IBM WebSphere XML Document Management Server Component. The interface is used to provide subscribers notification when a group definition is modified in an XCAP document which they own, or have sufficient rights to view or modify.

Overview

The SIP SUBSCRIBE and NOTIFY interface is described in the draft *RFC 3265 Session Initiation Protocol (SIP)-Specific Event Notification*. The IBM XDMS implementation of subscription and notification is based on the specifications described in the *XML Document Manager Specification OMA-TS-XDM_Core-V1_0_1* and *OMA-TS-XDM_Core-V2_0* for the new xcap-diff event type.

Subscription types

The IBM XDMS supports two type of subscription events:

UA-profile events
UA-profile events allow for subscription to XDMS documents. Notification patch operations may include a <changelog> element describing changes made to the subscribed document.

xcap-diff events
xcap-diff events allow for subscriptions to documents as well as elements and attributes within those elements. These documents, elements, or attributes may already exist, or if not, subscriptions can be made in pending state. xcap-diff events also allow for batch subscriptions, where more than one resource is subscribed to in one session. Notifications may include patch operations for document level subscriptions, and complete element or attribute content for node level subscriptions.

Subscription notifications

The SIP NOTIFY method is used to notify a SIP client (also known as an end point) about an event change or changes to a document that the SIP application had earlier requested notification about through an earlier SIP SUBSCRIBE request. In the case of IBM XDMS, the SIP NOTIFY is sent when a change is made to an XCAP document that has been subscribed to. The SIP NOTIFY generated contains an xcap-diff document when the initial SIP SUBSCRIBE has the header accept: application/xcap-diff+xml.

The following conditions will cause a SIP NOTIFY request to be sent:

- If SIP SUBSCRIBE request results in a successful 200 response code then a SIP NOTIFY request is sent to the requester with the state of the subscription. If the response to the SIP SUBSCRIBE is not a 200 response code then no SIP NOTIFY is sent
- If the subscribed document has been modified or deleted a SIP NOTIFY is sent to the SIP client.

Subscription-State headers provided in the SIP NOTIFY requests include the following:

- ACTIVE: indicates that the subscription has been accepted.
- TERMINATED: the subscription is not active. A `reason` tag will be included with the following information:
  - Timeout: The subscription expired or was canceled.
  - Noresource: The document which was subscribed to has been deleted.

Restrictions

1. In IBM XDMS when using UA-profile events XCAP SUBSCRIBE is only performed against an XCAP XML document. It is important to understand that the interface does not allow a subscribe request against a resource element or attribute.
2. Because the xcap-caps is a generated document, the SUBSCRIBE interface does not accept subscriptions to it.
3. The directory.xml file can only be subscribed to in Access Control List (ACLS) AUIDs. This file does not exist in standard AUIDs and is not available for subscriptions.

Security

The XCAP interface supports authentication to ensure that a requester has the appropriate privilege to perform the requested operation.

In order to issue a SIP SUBSCRIBE request, the requester must have read permission on the target document being subscribed to. If no read permission has been granted then a `403 Forbidden` response will be returned.

The p-asserted-identity header field is used to contain the identity of the user sending a SIP message that will be used for verify by authentication. The asserted identity header is parsed by the WebSphere IMS Connector Trust Association Interceptor (TAI). Refer to the TAI topics in this information center for more information about how to configure the TAI for handling asserted identities.

SUBSCRIBE headers

A SIP application can issue a SUBSCRIBE request for notification of a future change or changes to a document. The XCAP XML document header information for the SUBSCRIBE request is checked by the XCAP server to determine whether the request should be honored or rejected.

The following is a sample SUBSCRIBE request code snippet showing the format of SUBSCRIBE request. The event header consists of specific tags, separated by a semi-colon, which provide detailed information for the SUBSCRIBE request.

Accept header

The accept header parameter must be *application/xcap-diff+xml* if you if you want to receive a xcap-diff document with your notifies. This is the case with both **UA-profile** and **xcap-diff** events.

UA-profile event header

When IBM XDMS receives a subscribe request it scans the document header for the required **UA-profile** or **xcap-diff** event. If the Event header format is incorrect, a `489 Bad Event` response will be returned

ua-profile tag
The event for a UA-profile event subscription.

profile-type tag
The profile-type tag is mandatory and must be set to *application* as defined by the SIP UA-profile specification.

vendor tag
The vendor tag is mandatory. IBM XDMS ensures that the tag exists, but it can be set to any value.

model tag
The model tag is mandatory. IBM XDMS ensures that the tag exists, but it can be set to any value.

version tag
The version tag is mandatory. IBM XDMS ensures that the tag exists, but it can be set to any value.

auid tag

The auid tag specifies the AUID of the target document. It is not required if you specify the AUID in the document tag

document tag

The value of the document tag is used to identify the document that is being subscribed to.

- If the auid tag and document tag do not exist the request URI in the TO header of the SUBSCRIBE request is used. This URI should represent a service URI in a rls-services document. The rls-services definition is retrieved and the associated resource-lists XCAP URI is used for the subscription.
- If the service URI does not exist in an rls-services document, or if the document represented by the XCAP URI associated with that service URI does not exist in IBM XDMS, a 404 Not Found response will be returned.
- If the auid tag is rls-services and the document tag is "global/index", then the rls-services is again used to retrieve the associated resource-lists XCAP URI used for the subscription. However, the Request-URI is used to match the service URI instead of the TO header.
- If the auid tag is specified but the document header is not specified, then a user home directory for that AUID is being subscribed to. The XUI for the user's home directory is specified in the Request-URI.
- The document URI may be in the form of:
  - AUID/users/XUI/document
  - AUID/global/document
  - users/XUI/document
  - global/document

  In order to process the document URI, IBM XDMS checks to see if the document URI starts with either `global` or `users`. If it does, IBM XDMS requires an AUID, which must be specified using the auid tag

Sample ua-profile event header

```
Event: ua-profile;profile-type="application";vendor="ibm";model="xdms";version="7.0";
auid="resource-lists";document="users/sip:joe@us.example.com/mydocument.xml"
```

xcap-diff event header

xcap-diff tag

This event is for xcap-diff SIP subscribe event which allows for node level subscriptions.

diff-processing tag

The diff-processing parameter has three options:

xcap-patching

Provides xcap diff patch operations, this is default value assigned if the **diff-processing** parameter is not included in the subscribe request.

no-patching

Patch operations are not provided.

aggregate

This value will perform the same as xcap-patching in this 7.0 release of XDMS

Sample xcap-diff event header

```
Event: xcap-diff;diff-processing=xcap-patching
```

Expires header

The `Expires` header is an optional value for IBM XDMS. The value of this header field specifies the time in seconds that the subscription request is valid.

- If the `Expires` header does not exist, the value from the `subscribeExpiresDefault` configuration parameter is used.
- If the `Expires` value is zero the subscription is cancelled. Even if a subscription does not exist in IBM XDMS, a *200 OK* response will be returned to the requester. This will trigger a NOTIFY with a `Subscription-State` of `terminated`.
- If the `Expires` value is non-zero, the value will be checked against the `Min-Expires` and `Max-Expires` values.

  The actual expiration time values are set using a minute boundary, which means the time value after being read into the system is then reduced to the next lower minute value. For example, if the `Expires` time value was 80 seconds the next lower minute value of 60 seconds will be used. If the time value was 170 seconds the next lower minute value would be 120 seconds.
- If the `Expires` value is less than zero, the value is considered malformed and is set to the default value.
- If the `Expires` value is greater than zero, but less than the minimum time defined in the configuration property `subscribeExpireMin`, a *423 Interval Too Brief* response will be returned.
- If the `Expires` value is greater than the configured maximum value as defined in the configuration property `subscibeExpireMax` it is set to the maximum value defined in the property `subscibeExpireMax`.

Header return codes

Table 17. Event header return codes

Code	Description
200 OK	The SUBSCRIBE request was accepted and a subscription has been created or updated. A NOTIFY response is returned to the requester with the duration of the initial connection (unless the request is being refreshed). If the response class is not a 200 level response, the NOTIFY response will be a final response indicating there will be no further NOTIFY messages. If a subscription already exists for the group, and the intended endpoint of the notification is specified in the SUBSCRIBE request, the subscription will be updated.
403 Forbidden	The SUBSCRIBE request failed due to an authorization failure.
404 Not Found	The document specified in the header was not found at the specified URI.
423 Interval Too Brief	The duration in the SUBSCRIBE message was too short.
489 Bad Event	The Event header format is incorrect.
500 Internal Server Error	The SUBSCRIBE request failed due to some internal error (not header related).

UA-profile examples

The following are examples of subscribe and notify events using UA-profile:

UA-profile event subscription example

```
SUBSCRIBE sip:service@1.2.3.4:5060 SIP/2.0
Via: SIP/2.0/TCP 5.6.7.8:5060
From: <sip:user1@5.6.7.8:5060>;tag=1
To: <sip:john.doe@us.example.com>
Call-ID: 1.4192.3.4.5.6@call.id
Event: ua-profile;profile-type="application"
Max-Forwards: 70
CSeq: 1 SUBSCRIBE
P-Asserted-Identity: "John Doe" <sip:john.doe@us.example.com>
Privacy: none
Expires: 3600
Accept: application/xcap-diff+xml
Contact: sip:user1@5.6.7.8:5060
Content-Length: 0
```

UA-profile event subscription notifications

Change notification for UA-profile events are enabled by the *sendUAProfileChangeLog* resource environment provider, which is disabled by default. UA-profile events subscriptions and notifications are limited to XDMS documents and any change in a document will result in a notification.

Typical UA-profile event subscription notifications will include the following attributes for the resource:

- New e-tag
- Previous e-tag
- Change log `<changelog>`

The change log contains the patch operations, for example:

```
<add sel="resource-lists/list"><entry uri="buddy1@us.example.com"></entry></add>
```

The following example shows notification resulting from a UA-profile subscription with change notification:

```
NOTIFY sip:user1@5.6.7.8:5060 SIP/2.0
 Event: ua-profile;profile-type="application";vendor="ibm";model="xdms";version="7.0";
auid="resource-lists";document="users/sip:joe@us.example.com/mydocument.xml"
From: <sip:john.doe@us.example.com>;tag=6211624529223376_local.1149543099406_2_2
To: <sip:user1@5.6.7.8:5060>;tag=1
Call-ID: 11.4192.3.4.5.6@call.id
Max-Forwards: 70
CSeq: 1 NOTIFY
Content-Type: application/xcap-diff+xml
Content-Length: 285
Via: SIP/2.0/TCP 1.2.3.4:5060;branch=z9hG4bK655017094028370
Contact: <sip:1.2.3.4:5060;transport=tcp>
Subscription-State: active

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
 xcap-root="http://MyHost:9080/services/">
 <document new-etag="MTE0OTU0MzIxNjkxNQ=="
doc-selector="resource-lists/users/sip:john.doe@us.example.com/MyBuddies.xml"
previous-etag="MTE0OTU0MzIxNjkxNQ==">
  <change-log>
   <add sel="resource-lists/list"><entry uri="buddy1@us.example.com"></entry></add>
  </change-log>
 </document>
</xcap-diff>
```

SUBSCRIBE flow using rls-services documents

The way in which IBM XDMS processes a SUBSCRIBE request depends on what is found in the Event header. If the document tag exists in the Event header of the SUBSCRIBE request, then the process flow is simple. However, if no auid tag and no document tag exists in the Event header, then IBM XDMS assumes the subscription will use an rls-services document flow. The following topic gives an example of how to convert the SIP URI provided in the To header to that of a resource-list referenced by a service URI defined in a rls-services document.

If the Event header in the SUBSCRIBE request does not contain an auid tag and document tag, then IBM XDMS assumes that the subscription is based upon the SIP URI provided in the To header:

```
To: <sip:john.doe@us.example.com>
```

In order to be able to return an XCAP URI in the NOTIFY response, a mapping is required from a user-defined SIP URI, to an XCAP URI of a document stored in IBM XDMS. This mapping is provided through the use of the rls-services document. .

In order for a subscription to complete properly, there are multiple documents and document elements that must be created properly to reference one another. The SIP URI provided in the To header of the SUBSCRIBE request must be defined in a URI attribute of a `<service>` element of an rls-services document. That `<service>` element must contain a resource-list element which contains the XCAP URI that was used to create the resource-lists document (group definition) that is being subscribed to.

The following information shows the required process to create the appropriate documents for a subscription that provides only a SIP URI::

Create the resource-lists document

1. Create a resource-lists document containing your group list.

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
    <list name="friends">
        <entry uri="sip:friend1@example.com">
            <display-name>My Friend1</display-name>
        </entry>
        <entry uri="sip:friend2@example.com">
            <display-name>My Friend2</display-name>
        </entry>
    </list>
</resource-lists>
```

2. Store the document in IBM XDMS using an XCAP PUT request. The URL used for the PUT should be something like the following:

```
http://MyHost:9080/services/resource-lists/users/sip:john.doe@us.example.com/MyBuddies.xml
```

Create the rls-services document

1. Create an rls-services document that references the ″friends″ list within the `MyBuddies.xml` document which was just stored in the previous step.

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services xmlns:rl="urn:ietf:params:xml:ns:resource-lists"
         xmlns="urn:ietf:params:xml:ns:rls-services">
   <service uri="sip:john.doe@us.example.com">
      <resource-list>
          http://MyHost:9080/services/resource-lists/users/sip:john.doe@us.example.com/MyBuddies.
      </resource-list>
```

```
      <packages>
         <package>presence</package>
      </packages>
   </service>
</rls-services>
```

2. Be sure to check that the following sections of the rls-services document are coded correctly:
   - The document selector portion of the XCAP URI (everything before the /~~/ separator) specified in the resource-list element is exactly the same as it was used to create your resource lists. The XCAP spec requires this to point to a list element within that document which is why the XPath extension was added to that URI. Because a subscription is against a document rather than a list, the XPath extension will be ignored for subscription, but it must be there to be a valid rls-services document.
   - The `uri` tag specified in the service element is the SIP URI which is to be used in the To header of the SUBSCRIBE request.
   - When you store this rls-services document in IBM XDMS, the tree/XUI portion of the XCAP URI that is used to store the rls-services document must be the same as the tree/XUI portion of the XCAP URI that is specified in the resource-list element. In this case, `users/sip:john.doe@us.example.com` would be the tree/XUI.

     For this example, the URL used for the PUT of the rls-services document should be something like the following:

     `http://MyHost:9080/services/rls-services/users/sip:john.doe@us.example.com/MyBuddies.xml`

Create the SUBSCRIBE Request

After the resource-lists and rls-services documents are stored in IBM XDMS, it is possible to perform a SUBSCRIBE request as follows:

```
SUBSCRIBE sip:service@1.2.3.4:5060 SIP/2.0
Via: SIP/2.0/TCP 5.6.7.8:5060
From: <sip:user1@5.6.7.8:5060>;tag=1
To: <sip:john.doe@us.example.com>
Call-ID: 1.4192.3.4.5.6@call.id
Event: ua-profile;profile-type="application"
Max-Forwards: 70
CSeq: 1 SUBSCRIBE
P-Asserted-Identity: "John Doe" <sip:john.doe@us.example.com>
Privacy: none
Expires: 3600
Accept: application/xcap-diff+xml
Contact: sip:user1@5.6.7.8:5060
Content-Length: 0
```

NOTIFY response

Here is an example of the NOTIFY response from the previous SUBSCRIBE request:

```
NOTIFY sip:user1@5.6.7.8:5060 SIP/2.0
Event: ua-profile;profile-type="application";vendor="ibm";model="xdms";version="7.0";
9 auid="resource-lists";document="users/sip:joe@us.example.com/mydocument.xml"

From: <sip:john.doe@us.example.com>;tag=6211624529223376_local.1149543099406_2_2
To: <sip:user1@5.6.7.8:5060>;tag=1
Call-ID: 11.4192.3.4.5.6@call.id
Max-Forwards: 70
CSeq: 1 NOTIFY
Content-Type: application/xcap-diff+xml
Content-Length: 285
```

```
Via: SIP/2.0/TCP 1.2.3.4:5060;branch=z9hG4bK655017094028370
Contact: <sip:1.2.3.4:5060;transport=tcp>
Subscription-State: active

<?xml version="1.0" encoding="UTF-8"?>
     <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
         xcap-root="http://MyHost:9080/services/">
     <document new-etag="MTE0OTU0MzIxNjkxNQ=="
doc-selector="resource-lists/users/sip:john.doe@us.example.com/MyBuddies.xml"
previous-etag="MTE0OTU0MzIxNjkxNQ==""/>
     </xcap-diff>
```

Verification

If you successfully created the resource-lists and rls-services document, and correctly coded the SUBSCRIBE request, you should receive a 200 response from NOTIFY. If you do not receive a 200 response you may want to examine some of the following:

- If the URI specified in the To header is not found in any rls-services document, a `404 Not found` response is returned in response to the SUBSRCIBE request. Ensure that the URI defined in the rls-services `<service>` element is the same as the URI defined To header. The referenced `<resource-list>` element must also refer to an existing resource-lists document.
- If the `service` element contains a `list` element rather than a `resource-list` element then a `404 Not Found` response is returned. IBM XDMS does not allow a requester to subscribe directly to a local list.
- If the URI specified in the `resource-list` element does not point to a valid document stored within IBM XDMS, a `404 Not Found` response is returned.
- The `xcap-root` value used in the xcap-diff is the configuration parameter defined for the global ibm-xdms Resource Environment Provider property `xcapRoot`. In a production environment the `xcapRoot` property should be the address of the Aggregation Proxy or edge proxy server depending on which server the XCAP client is configured to communicate directly with.

xcap-diff examples

The following are examples of xcap-dff subscription events:

Notification for subscriptions using the xcap-diff event vary depending on the level of subscription. Subscriptions to documents can receive patch operations. The `changelog` element is not used in xcap-diff events; instead, the patch operations appear in the `xcap-diff` element. Notifications for subscriptions using xcap-diff events at the element and attribute levels do not return patch operations; instead, the patch operations appear in the document element.

Note: To enable subscription and notification at the element or attribute level, node level constraints must be defined in the AUID descriptor files. Information on customizing the AUID descriptor files is available in the IMS Enablement Toolkit information center.

About node level constraints

Subscriptions to document elements and attributes must follow the defined node level constraints for each application. Attributes are not defined in the node level constraints, but can be subscribed to if they are appended to a matched element node level. The applications that are shipped with IBM XDMS have the following default constraints:

Note: *$1* is a wildcard variable.

resource-lists:
resource-lists/list[@name="$1"]

rls-services
rls-services/service[@uri="$1"]

pres-rules
ruleset/rule[@id="$1"]

group-usage-list
resource-lists/list[@name="$1"]

acls ruleset/rule[@id="$1"]

groups
group/list-service[@uri="$1"]

user-profile
user-profile/user-profiles[@uri="$1"]

locked user-profile
user-profile/user-profiles[@uri="$1"]

access-rules
ruleset/rule[@id="$1"]

xcap-diff event document- or directory-level notifications

xcap-diff subscription notifications at the document or directory level are similar to UA-profile events. Each receives back the new and previous e-tag for the document and patch operations. The xcap-diff event does not use the change log. Instead, patch operations at the document level might look like this:

```
 <document new-etag="MTE0OTU0MzIxNjkxNQ==" doc-selector="resource-lists/users/sip:john.doe@us.exam

<add sel="resource-lists/list"><entry uri="buddy1@us.example.com"></entry></add>

 </document>
```

xcap-diff event element- or attribute-level descriptions

For xcap-diff subscriptions at the node level, either element or attribute, no patch operations are provided. Instead, the content is provided. For example, if you were subscribed to a list in a document with the name attribute equal to *friends*, the notification for that element may look like this:

```
<element sel="resource-lists/global/doc.xml/~~/resource-lists/list[@name="friends"]" exists="true"
 <list name="friends">
    <display-name>Golf Friends</display-name>
    <entry uri="sip:tiger.woods@example.com">
     <display-name>Tiger Woods</display-name>
     </entry>
     <entry uri="sip:vijay.singh@example.com">
        <display-name>Vijay Singh</display-name>
     </entry>
   </list>
</element>
```

The following is an example of an initial notification following a subscribe, showing all the resources you are subscribed to. Subsequent notifies will contain only one resource per notify.

```
NOTIFY sip:user1@example.com:5060;transport=UDP
Subscription-State: active;expires=120
Event: xcap-diff;diff-processing=xcap-patching
Content-Type: application/xcap-diff+xml

<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff" xcap-root="localhost:9080/services">
 <document sel="resource-lists/users/sip:xdms1@us.acme.com/doc1.xml" new-etag="123213234"/>
 <document sel="resource-lists/users/sip:xdms1@us.acme.com/doc2.xml" new-etag="434543533"/>
 <document sel="rls-services/users/sip:xdms1@us.acme.com/index" new-etag="454355765"/>
 <element sel="resource-lists/global/doc.xml/~~/resource-lists/list[@name="friends"]" exists="true"/>
  <list name="friends">
   <display-name>Golf Friends</display-name>
    <entry uri="sip:tiger.woods@example.com">
     <display-name>Tiger Woods</display-name>
    </entry>
    <entry uri="sip:vijay.singth@example.com">
     <display-name>Vijay Singh</display-name>
    </entry>
  </list>
 </element>
 <attribute sel="resource-lists/global/doc.xml/~~/resource-lists/list[@name="buddies"]/@id" exists="
</xcap-diff>
```

SUBSCRIBE flow for xcap-diff using rls-services documents

The xcap-diff event header subscription processing differs from the UA-profile event header.

About this task

The following describes a possible flow of events during an xcap-diff subscription event:

1. A user sends a subscribe request
2. Subscription request entries are parsed
3. The user receives 200 response, then initial notify
4. User updates one of entries in one of the documents from the subscribed directory.
5. Subscribed user receives a notify
6. User receives one termination notify

Following is a detailed description of these events.

1. A user sends a subscribe request.

```
SUBSCRIBE sip:user1@example.com:5080;transport=UDP SIP/2.0
      Via: SIP/2.0/UDP example.com:5060
      From: <sip:user1@example.com:5060>
      To: <sip:tester1@example.com>
      Call-ID: 1
      Max-Forwards: 70
      Expires: 120
      Event: xcap-diff;diff-processing=xcap-patching
      accept: application/xcap-diff+xml
      CSeq: 1 SUBSCRIBE
      Contact: sip:user1@example.com:5060;transport=UDP
      Subject: UAS
      Content-type: application/resource-lists+xml
      Content-Length: 256

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
    <list>
      <entry uri="resource-lists/users/sip:xdms1@us.acme.com"/>
```

```
      <entry uri="rls-services/users/sip:xdms1@us.acme.com/index"/>
      <entry uri="resource-lists/global/doc.xml/~~/resource-lists/list[@name="friends"]"/>
      <entry uri="resource-lists/global/doc.xml/~~/resource-lists/list[@name="buddies"]/@id"/>
      <entry uri="resource-lists/users/sip:xdms2@us.acme.com/doc.xml"/>
      <entry uri="bad_auid/global/dumb.xml"/>
    </list>
</resource-lists>
```

2. Entries are parsed and sent through the filter chain for validation, authorization, and subscription addition.
   a. Entries 1 and 2 are valid and exist.
   b. Entries 3 and 4 are valid, exist, and match the node-selector template for their AUID.
   c. Entry 5 does not exist yet.
   d. Entry 6 is not valid.
3. User receives a 200 response, then the initial notify.
   a. Document elements 1 and 2 were the only ones currently in the user directory.
   b. Document element 3 is from the second entry in the subscribe.
   c. Element element is from the third entry in subscribe, and shows contents.
   d. Attribute element is from the fourth entry in the subscribe, and shows value.
   e. Entry 5 from subscribe is in pending state and subscribed to, but not in notify.
   f. Entry 6 from subscribe was removed because it was not valid.

```
NOTIFY sip:user1@example.com:5060;transport=UDP
Subscription-State: active;expires=120
Event: xcap-diff;diff-processing=xcap-patching
Content-Type: application/xcap-diff+xml
<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
     xcap-root="localhost:9080/services">

    <document sel="resource-lists/users/sip:xdms1@us.acme.com/doc1.xml"
              new-etag="123213234"/>
   <document sel="resource-lists/users/sip:xdms1@us.acme.com/doc2.xml"
              new-etag="434543533"/>
    <document sel="rls-services/users/sip:xdms1@us.acme.com/index"
              new-etag="454355765"/
    <element sel="resource-lists/global/doc.xml/~~/resource-lists/list[@name="friends"]"
             exists="true"/>
       <list name="friends">
          <display-name>Golf Friends</display-name>
          <entry uri="sip:tiger.woods@example.com">
            <display-name>Tiger Woods</display-name>
          </entry>
          <entry uri="sip:vijay.singth@example.com">
            <display-name>Vijay Singh</display-name>
          </entry>
       </list>
    </element>
    <attribute sel="resource-lists/global/doc.xml/~~/resource-lists/list[@name="buddies"]/@id"
               exists="true"/>5</attribute>
</xcap-diff>
```

4. User updates one of entries in one of the documents from the subscribed directory.

a. The user sends an xcap put to the following document selector: `resource-lists/users/sip:xdms1@us.acme.com/doc1.xml/~~/resource-lists/list[@name="friends"]/entry[@uri="steve"]` and the following content:

```
<entry uri="steve">
    <display-name>Stephen</display-name>
   </entry>
```

5. The user receives a notification.

```
NOTIFY sip:user1@example.com:5060;transport=UDP
Subscription-State: active;expires=120
Event: xcap-diff;diff-processing=xcap-patching
Content-Type: application/xcap-diff+xml
<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
           xcap-root="localhost:9080/services">
   <document sel="resource-lists/users/sip:xdms1@us.acme.com/index"
             prev-etag="3413415" new-etag="4154351">
      <replace sel="resourcelists/list[@name="friends"]
                  /entry[@uri="steve"]">
         <entry uri="steve">
            <display-name>Stephen</display-name>
         </entry>
       </replace>
    </document>
</xcap-diff>
```

6. User receives one termination notify.

```
NOTIFY sip:user1@example.com:5060;transport=UDP
Subscription-State: terminated
Event: xcap-diff;diff-processing=xcap-patching
Content-Type: application/xcap-diff+xml
```

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. (Omit the password if security is not enabled.)

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)**. To ensure the accuracy of your results, select the check box labeled **Use sequential counter updates**.
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 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. (Omit the password if security is not enabled.)

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.

The following set of metrics (indicators and counters) is supported for each of the IBM XDMS applications (AUIDs).

Table 18. Performance metrics for AUIDs

Variable	Description
PUT Requests	Number of PUT requests
GET Requests	Number of GET requests
DELETE Requests	Number of DELETE requests
POST Requests	Number of POST requests
SUBSCRIBE Requests	Number of SUBSCRIBE requests
Success rate	Percentage of successful responses (number of requests that receive a 2XX response code, divided by the total number of requests)

Table 18. Performance metrics for AUIDs (continued)

Variable	Description
NOTIFIES	Number of notifies attempted. A NOTIFIES counter is available for each of the AUIDs in the XDMS enablers. This counter is incremented for all of the following events: • Initial notify in response to a SUBSCRIBE, UNSUBSCRIBE, or RESUBSCRIBE request • Notify generated after a document is modified • Notify generated when a subscription has expired For UA-profile event subscriptions, the NOTIFY count is updated under the AUID of the documents in the subscription. However, for xcap-diff event subscriptions, the NOTIFY count is updated under the sip-xcap-diff-event AUID–not under the AUID of the documents in the subscription.
PUT Latency	Average latency time for PUT requests
GET Latency	Average latency time for GET requests.
DELETE Latency	Average latency time for DELETE requests
POST Latency	Average latency time for POST requests
SUBSCRIBE Latency	Average latency time for SUBSCRIBE requests

The following metrics are supported for the Aggregation Proxy.

Table 19. Performance metrics for the Aggregation Proxy

Name	Description
PUT Requests	Number of PUT requests
GET Requests	Number of GET requests
DELETE Requests	Number of DELETE requests
PUT Latency	The average latency time for PUT requests
GET Latency	The average latency time for GET requests
DELETE Latency	The average latency time for DELETE requests

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 of the IBM WebSphere software for Telecom products.

Compatibility

This release of IBM WebSphere XML Document Management Server Component 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 XDMS. Review the ITLM documentation carefully to determine which operating systems, maintenance levels, and Linux kernel versions are supported.

Installation

During the installation of IBM XDMS, 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 IBM XDMS EAR into WebSphere Application Server. It is located within the EAR file in the TIVREADY subdirectory, and its name is IMSXDM0603.SYS2.

License file

IBM XDMS 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 component as a J2EE product.

Chapter 5. Troubleshooting IBM XDMS

Logs store information that can help you troubleshoot problems that might occur as you install, configure, and use IBM XDMS.

Using ISA 4.0 add-ons to communicate with IBM Support

To help you communicate with IBM Support, an IBM Support Assistant (ISA) 4.0 product add-on is available on the Web for IBM WebSphere XML Document Management Server Component. 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.

Audit logging

It is possible to log HTTP and SIP requests from specific users and/or roles. The audit logger will log HTTP PUT, GET, DELETE, POST requests and SIP SUBSCRIBE requests.

About this task

To specify the users/roles to log, change the REPs *auditLogByUser* and *auditLogByRole* for the AUID you want to monitor.

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. (Omit the password if security is not enabled.)

c. Click **Log in**.

2. Click **Resources** → **Resource Environment** → **Resource Environment Providers**.
3. Click ***AUID*** → **Custom Properties**.
4. Click *auditLogByUser* and change values to desired value.

Option	Description
auditLogByUser	Default is empty. Comma-separated list of users. If specified, then log requests and responses onlyfor these users. Specifying '*' logs all users.

Results

The output is logged in the `trace.log` for the server of the specified AUID. The `trace.log` can be found at:

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`

What to do next

To enable trace, refer to the Trace Logging section of this Information Center. The required audit logging classes are

com.ibm.xmds.sip.filter.impl.AuditLoggerSipFilter

com.ibm.xmds.xcap.filter.impl.AuditLoggerXcapFilter

The audit logging trace levels are:

Fine Logs the HTTP method, XCAP URI, principal making the request, and the response code going back..

Finer Logs HTTP headers for both the request and response, along with the elements in Fine tracing.

Finest Logs the content from the request and content in the returned response, along with the elements in Finer tracing.

Monitoring log messages

IBM WebSphere XML Document Management 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 XML Document Management Server Component logs its messages to the WebSphere Application Server JVM log (`SystemOut.log`) and its trace messages to the WebSphere Application Server trace log (`trace.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:

[3/14/09 10:45:28:004 CST] 00000036 GroupListImpl E GLSR0006E: Unable to connect to the repository adapter web service endpoint for the following reason: null

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. (Omit the password if security is not enabled.)
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*.)

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 XML Document Management 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. (Omit the password if security is not enabled.)
   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 XML Document Management Server Component, use `com.ibm.xdms.*=all` as a logger group name.

   com.ibm.xcap.*

   com.ibm.xdms.agp.*

   com.ibm.xdms.common.*

   com.ibm.xdms.function.*

   com.ibm.xdms.install.*

   com.ibm.xdms.osgi.*

   com.ibm.xdms.sip.*

   com.ibm.xdms.utils.*

   com.ibm.xdms.xcap.*
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 20. TAI trace groups and trace loggers

Trace group	Trace loggers
com.ibm.imsconnector.tai.*	*BaseIMSInterceptor *HttpInterceptor *SipInterceptor

To control the trace level for the IBM XDMS component and its subcomponents, use the options on the `com.ibm.xdms.*` and `com.ibm.xcap.*` trace group. For more specific levels of tracing, use the following trace groups, which are relevant to IBM XDMS:

Table 21. XDMS trace group and trace loggers

Trace group	Trace loggers
com.ibm.glm.http.security.*	none
com.ibm.xcap.*	*client.* *xml.*
com.ibm.xdms.agp.*	*common.* *exception.* *install.* *servlet.*
com.ibm.xdms.*	*common.* *common.impl.* *exception.* *exception.sip.* *exception.xcap.* *function.* *function.impl.* *install.* *osgi.* *utils.* *version.*
com.ibm.xdms.sip.*	*filter.* *filter.impl.* *impl.* *servlet.* *utils.*
com.ibm.xdms.xcap.*	*filter.* *filter.impl.* *impl.* *servlet.*

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.

Faults and alarms

IBM XDMS provides notification of application-critical events through the WebSphere Telecom Web Services Server Faults and Alarm notification service.

Within the ibm-xdms Resource Environment Provider for each IBM XDMS application, the entry `alarmInterval` allows the system administrator to declare the interval on which to receive duplicate alarms. The default value for this entry is 15 minutes.

An alarm is logged for any of the following events:

- Initialization of a server fails.
- DB2 is not accessible for accessing document AUID tables or usage records.
- JMS is not accessible to log notification events.
- The Aggregation Proxy cannot connect with a configured IBM XDMS.
- The Aggregation Proxy is configured with more than one IBM XDMS supporting the same AUID for the same domain.
- IBM XDMS cannot access Resource Environment Providers for configuration settings.

A fault is logged for any of the following events:

- An `<external>` list cannot be resolved during an access control check from an authorization policy.
- An IBM XDMS server shuts down (a fault for each of the AUIDs served by IBM XDMS is logged).

Chapter 6. Reference

Standards and specifications

The IBM WebSphere XML Document Management Server is primarily based on the OMA XDM Core specification. However, the OMA XDM Core specification references other specifications both from other OMA specifications and IETF specifications. The following tables list the standards organizations and the various specifications used in building the IBM implementation.

Open Mobile Alliance

http://www.openmobilealliance.org/

Table 22. OMA specifications

Specification title	Version	Notes
OMA-TS-XDM_Core	1.0.1	Full compliance
OMA-TS-XDM_Core	2.0	Partial implementation for XDMS POST for search. No Search Proxy.
OMA-TS-XDM_Shared_List	1.0.1	Implementation for URI List and Group Usage List
OMA-TS-XDM_Shared_Profile	1.0	Full compliance
OMA-TS-XDM_Shared_Policy	1.0	Full compliance
OMA-TS-XDM_Shared_Group	1.0	Full compliance
OMA-TS-Presence_SIMPLE_XDM	1.0.1	Full compliance

IETF

http://www.ietf.org/

Table 23. IETF specifications

Specification title	Version	Notes
ietf-simple-xcap-list-usage	05	Full compliance
ietf-geopriv-common-policy	11	Full compliance
ietf-simple-xcap	12	Full compliance
ietf-sip-xcap-config	00	Full compliance
ietf-simple-xcap-diff	04	Partial implementation. No patch operations provided.
ietf-sipping-config-framework	09	Full compliance

Resource Environment Providers for the XDMS enablers

Several Resource Environment Providers (REPs) exist for the XDMS enablers.

Resource Environment Providers for each enabler

Table 24. Enabler Resource Environment Providers. Each enabler has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
superAdminUser	*superadmin*	String	Required	A user configured with the super-admin role (recommended value: superadmin) Can be changed dynamically at runtime
superAdminPassword	*password*	String	Required	Password for the super-admin role (recommended value: none) Can be dynamically changed at runtime
xcapRoot	*http://localhost:9080/services*	String	Required	The XCAP Root of the Aggregation Proxy, if installed, or the XCAP Root of the local server (recommended value: http(s)://*hostname*:*ports*/services) Can be changed dynamically at runtime
preserveWhitespace	*false*	Boolean	Required	Whether the DataStore function should preserve whitespace in the XML documents (recommended value: true) Can be changed dynamically at runtime
alarmInterval	*15*	Integer	Not required	The number of minutes between Alarm notifications for the same alarm (recommended value: 15) Can be changed dynamically at runtime

Table 24. Enabler Resource Environment Providers (continued). Each enabler has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
includeNotifyTransaction	*false*	Boolean	Required	If an error occurs during JMS notification (system error), this Boolean flag determines whether the system error causes the database transaction to roll back database updates. If true, the JMS error causes a rollback and if false, it has no effect on the database transaction (recommended value: false) Cannot be changed dynamically at runtime
enableDraftPutConditionalSupport	*false*	Boolean	Not required	Enables draft support of conditional headers (If-None-Match and If-Match) (recommended value: false, unless this support is needed) Draft versions of RFC 4825 allowed requests with conditional headers set to '*' to test for existence of a particular element/attribute of a document, rather than the document itself. Only applies if the request has a node selector and the headers only contain '*' Can be changed dynamically at runtime

Table 24. Enabler Resource Environment Providers (continued). Each enabler has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
repRefreshInterval	5	Integer	Not required	Refresh interval, in minutes, of REP custom properties (recommended value: 5 or more) For values that can be changed dynamically, the custom property information is read at this interval, allowing the configuration to be updated without a restart of the server. A value of zero will mean the information is never refreshed Cannot be changed dynamically at runtime
initialNotifyWaitTime	0	Integer	Not required	Milliseconds to pad between the sending out of a 200 response for a SUBSCRIBE request and the initial notify for that SUBSCRIBE request

Resource Environment Providers for each AUID

Table 25. Resource Environment Providers. Each AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
enableSchemaValidation	true	Boolean	Required	Enables or disables XML schema validation (recommended value: true) Can be changed dynamically at runtime
listMinimumMembers	-1	Integer	Not required	Minimum number of members in a list (recommended range: 0-10000) If not specified or a negative number, the minimum members policy is disabled Can be changed dynamically at runtime

Table 25. Resource Environment Providers (continued). Each AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
listMaximumMembers	7	Integer	Not required	Maximum number of members in a list (recommended range: 0-10000) If not specified or a negative number, the maximum members policy is disabled Can be changed dynamically at runtime
db!default!datasource	*jdbc/xdms*	String	Required	Data source used for communicating with the XDMS database (recommended value: jdbc/xdms) Cannot be changed dynamically at runtime The application must be restarted for this setting to take effect
db!default!table	*resourcelists*	String	Required	Database table to use for storing documents (recommended value: resourcelists) Cannot be changed dynamically at runtime The application must be restarted for this setting to take effect
enableStandardAuthorization	*true*	Boolean	Required	Enables or disables standard authorization (recommended value: true) Users with standard authorization have admin permission to their own home directory and read permission to global documents Can be changed dynamically at runtime

Table 25. Resource Environment Providers (continued). Each AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
authorizationPolicyAuid	*com.ibm.resource-lists-acls*	String	Required	AUID for authorization policy rules that represents ACLs for rls-services. (recommended value: com.ibm.resource-lists-acls) Cannot be changed dynamically at runtime
sipRespondPending	*false*	Boolean	Required	Whether to respond immediately to SIP SUBSCRIBE requests with a 202 status code meaning ″accepted but pending″ (recommended value: false) Can be changed dynamically at runtime
sipDefaultExpire	*3600*	Integer	Required	Default expire time in seconds if no expire time is specified in the SIP SUBSCRIBE request (recommended value: 3600) Can be changed dynamically at runtime
sipMinimumExpire	*1*	Integer	Not required	Minimum required expire time in seconds that is specified in the SIP SUBSCRIBE request (recommended range: 0-3600) If not specified or a negative number, the minimum expires policy is disabled Can be changed dynamically at runtime

Table 25. Resource Environment Providers (continued). Each AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
sipMaximumExpire	*1*	Integer	Not required	Maximum allowed expire time in seconds that is specified in the SIP SUBSCRIBE request (recommended range: 60-86400) If not specified or a negative number, the maximum expires policy is disabled Can be changed dynamically at runtime
usageRecordDatasource	*jdbc/xdmsur*	String	Not required	Data source used for communicating with the XDMSUR database that is used to store usage records (recommended value: jdbc/xdmsur) Cannot be changed dynamically at runtime The application must be restarted for this setting to take effect
enableXcapPutUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP PUT requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableXcapGetUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP GET requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime

Table 25. Resource Environment Providers (continued). Each AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
enableXcapDeleteUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP DELETE requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableXcapPostUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP POST requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableSipSubscribeUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for SIP SUBSCRIBE requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableExternalElementValidation	*false*	Boolean	Not required	Whether newly inserted external elements should be validated to ensure that there are no recursive references (recommended value: false) Can be changed dynamically at runtime
auditLogByRole	*super-admin*	String	Not required	Comma-separated list of users If specified, requests and responses are logged only for these users

Table 25. Resource Environment Providers (continued). Each AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
maxSearchResults	*0*	Integer	Not required	Number of search results to return from the query (default value: 0) If equal to or less than 0, all search results are returned Can be changed dynamically at runtime
sendUAProfileChangeLog	*false*	Boolean	Not required	Enables change log information within XCAP Diff of a UA Profile event notify

ACLS Resource Environment Providers

Table 26. ACLS Resource Environment Providers. Each ACLS AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
enableSchemaValidation	*true*	Boolean	Required	Enables or disables XML schema validation (recommended value: true) Can be changed dynamically at runtime
db!default!datasource	*jdbc/xdms*	String	Required	Data source used for communicating with the XDMS database (recommended value: jdbc/xdms) Cannot be changed dynamically at runtime The application must be restarted for this setting to take effect
db!default!table	*resourcelistsacls*	String	Required	Database table to use for storing documents (recommended value: resourcelistsacls) Cannot be changed dynamically at runtime The application must be restarted for this setting to take effect

Table 26. ACLS Resource Environment Providers (continued). Each ACLS AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
enableStandardAuthorization	*true*	Boolean	Required	Enables or disables standard authorization (recommended value: true) Users with standard authorization have admin permission to their own home directory and read permission to global documents Can be changed dynamically at runtime
authorizedAuid	*resource-lists*	String	Required	Specifies the AUID that these authorization policy rules represent (Recommended value: resource-lists) Cannot be changed dynamically at runtime
sipRespondPending	*false*	Boolean	Required	Whether to respond immediately to SIP SUBSCRIBE requests with a 202 status code meaning ″accepted but pending″ (recommended value: false) Can be changed dynamically at runtime
sipDefaultExpire	*3600*	Integer	Required	Default expire time in seconds if no expire time is specified in the SIP SUBSCRIBE request (recommended value: 3600) Can be changed dynamically at runtime

Table 26. ACLS Resource Environment Providers (continued). Each ACLS AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
sipMinimumExpire	*1*	Integer	Not required	Minimum required expire time in seconds that is specified in the SIP SUBSCRIBE request (recommended range: 0-3600) If not specified or a negative number, the minimum expires policy is disabled Can be changed dynamically at runtime
sipMaximumExpire	*1*	Integer	Not required	Maximum allowed expire time in seconds that is specified in the SIP SUBSCRIBE request (recommended range: 60-86400) If not specified or a negative number, the maximum expires policy is disabled Can be changed dynamically at runtime
usageRecordDatasource	*jdbc/xdmsur*	String	Not required	Data source used for communicating with the XDMSUR database that is used to store usage records (recommended value: jdbc/xdmsur) Cannot be changed dynamically at runtime The application must be restarted for this setting to take effect
enableXcapPutUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP PUT requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime

Table 26. ACLS Resource Environment Providers (continued). Each ACLS AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
enableXcapGetUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP GET requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableXcapDeleteUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP DELETE requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableXcapPostUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for XCAP POST requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
enableSipSubscribeUsageRecordLogging	*false*	Boolean	Not required	Enables or disables usage record logging for SIP SUBSCRIBE requests (recommended value: false) If enabled, the usageRecordDatasource must be configured Can be changed dynamically at runtime
auditLogByRole	*super-admin*	String	Not required	Comma-separated list of users If specified, requests and responses are logged only for these users

Table 26. ACLS Resource Environment Providers (continued). Each ACLS AUID has the following Resource Environment Provider properties:

Property name	Example value	Property type	Required?	Property description
maxSearchResults	*0*	Integer	Not required	Number of search results to return from the query (default value: 0) If equal to or less than 0, all search results are returned Can be changed dynamically at runtime
sendUAProfileChangeLog	*false*	Boolean	Not required	Enables change log information within XCAP Diff of a UA Profile event notify.

Example property

```
enableSchemaValidation=true
enableSchemaValidation.type=java.lang.Boolean
enableSchemaValidation.req=false
enableSchemaValidation.desc=Enables or disables XML schema validation [Recommended value: true].
```

Aggregation Proxy Resource Environment Providers

Several Resource Environment Providers exist for the Aggregation Proxy.

Aggregation Proxy Environment Provider properties

Table 27. Aggregation Proxy properties

Property name	Value	Type	Required?	Description
HTTPS_PROXY_XDMS	false	Boolean	yes	Whether or not this Aggregation Proxy's backend XDMS instances are accessed by an `https` URL
PROXY_ROOT	http://proxy.com:9082	string	yes	The portion of the Aggregation Proxy's URL ending with the context root
THREEGPP_IMS	false	Boolean	yes	Whether or not the 3GPP-GAA is present. If false, `Xdms-Asserted-Identity` is used
XCAP_CACHE_TIMEOUT	180	Integer	yes	Time in seconds that the XCAP-CAPS of each backend XDMS instance should remain in WebSphere's dynamic cache

Table 27. Aggregation Proxy properties (continued)

Property name	Value	Type	Required?	Description
superAdminUser	superadmin	String	yes	ID used by the Aggregation Proxy during initialization and for sending an initial XCAP-CAPS request to each backend XDMS instance
superAdminPassword	password	String	yes	Password used by the Aggregation Proxy during initialization
XDMS_URI	http://hostname1:9080/services #http://hostname2:9080/services	String	yes	XCAP root of XDMS, per domain
XDMS_URI!<domain>	http://sharedlist2.com:9080/services/resource-lists	String	yes	Defines the domain for which the list of XDMS enablers is supported
XDMS_URI_HTTPS	https://sharedlist1.com:9080/services #http://presrules1.com:9081/services	String	no	The XUIs of the backend XDMS clusters serviced by the Aggregation Proxy. List all XUIs here, separated by a # symbol
XDMS_URI_HTTPS !<domain>	https://sharedlist2.com:9080/services/resource-lists	String	no	Defines the secure domain for which the list of XDMS enablers is supported
alarmInterval	15	String	no	Time in minutes between alarm notifications for the same alarm (recommended value: 15)
BAD_XDMS_POLLING_INTERVAL	20	Integer	yes	Time in seconds for polling XDMS instances that have never been online
MAX_PROXY_THREAD_POOL_SIZE	60	Integer	yes	Maximum number of threads in the thread pool used to send proxy requests to XDMSes

Example Policy documents

Examples of XDMS Policy documents

External list

```
.
<ruleset>
      <rule id="read">
           <conditions>
              <external-list>
                     <entry anc="..xcapURItoList1.."/>
                     <entry anc="..xcapURItoList2.."/>
                  </external-list>
           </conditions>
           <actions/>
           <transformations/>
      </rule>
</ruleset>
```

Anonymous request

```
<ruleset>
      <rule id="read">
           <conditions>
                   <anonymous-request>
           </conditions>
           <actions/>
           <transformations/>
      </rule>
</ruleset>
```

Identity based authorization

```
 <ruleset>
       <rule id="read">
            <conditions>
                <identity>
                  <many domain="us.acme.com">
                      <except domain="rtp.us.acme.com"/>
                      <except id="sip:joe@us.acme.com"/>
                  </many>
                  <one id="sip:jane@tw.acme.com>
                </identity>
            </conditions>
            <actions/>
            <transformations/>
       </rule>
 </ruleset>
```

Example Resource List document

Example Resource List document. TestResourceList.xml

Resource List document

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists" xmlns:oau="urn:oma:xml:xdm:resource-
  <list name="friends">
    <list name="close-friends">
      <display-name>Close Friends</display-name>
      <external anchor="http://www.example.org/xcap/resource-lists/users/a/foo/~~/resource-lists/l
        <display-name>Marketing</display-name>
      </external>
      <entry uri="sip:joe@example.com">
        <display-name>Joe Smith</display-name>
      </entry>
      <entry uri="sip:nancy@example.com">
        <display-name>Nancy Gross</display-name>
      </entry>
```

```
    </list>
    <entry uri="sip:bill@example.com">
      <display-name>Bill Doe</display-name>
    </entry>
    <entry-ref ref="resource-lists/users/sip:bill@example.com/mylist/~~/resource-lists/list%5b@name=
  </list>
</resource-lists>
```

Chapter 7. 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 28. 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.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department number/Building number

Site mailing address
City, State; Zip Code
_U.S.A. (or appropriate country)

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

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:

DB2
IBM
pureXML
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 XML Document Management Server
IBM WebSphere XML Document Management Server
Version 7.0

Publication No. SC00-0000-00

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)

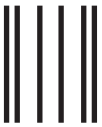
If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
SC00-0000-00

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

Part Number: 99F9999

Printed in USA

(1P) P/N: 99F9999

SC00-0000-00