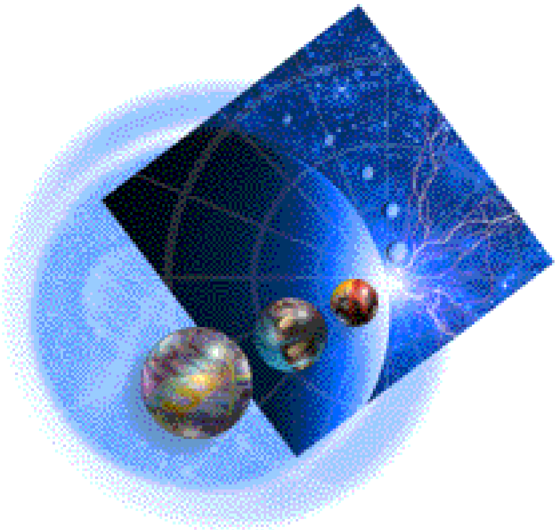
IBM WebSphere Everyplace Server, Service Provider Offering for Multiplatforms version 2.1



Voice Services

Administer

IBM WebSphere Everyplace Server, Service Provider Offering for Multiplatforms version 2.1

Why WebSphere Everyplace Server?

Become acquainted with installation models and deployment scenarios for WebSphere Everyplace Server.

WebSphere Everyplace Server components

Introduce yourself to all the WebSphere Everyplace Server components. There are feature, support, and third party components that make up WebSphere Everyplace Server.

Planning and installation

Learn about the installation program, Setup Manager, and review important information to help you plan and prepare to install WebSphere Everyplace Server.

System administration

Learn how to configure and adminster the various components of WebSphere Everyplace Server from one central location.

Reference...

- List of components
- Requirements and prerequisites
- Related Redbooks
- Default port numbers

About this information

About this documentation

The WebSphere Everyplace Server documentation discusses the planning, installation, and configuration of the IBM WebSphere Everyplace Server, Service Provider Offering for Multiplatforms version 2.1 (from now on referred to as WebSphere Everyplace Server or Everyplace Server).

Everyplace Server is an integrated, modular suite of software components that support connectivity of pervasive devices such as wireless phones, personal digital assistants (PDAs), and mobile computers, to online information.

To locate component specific documentation, select the desired component in the navigation tree of the InfoCenter.

Refer to the WebSphere Everyplace Server web site for the most recent product information.

Who should read this information

Everyplace Server documentation is intended for system administrators responsible for installing and configuring Everyplace Server components and for individuals using and administering Everyplace Server. All users and administrators should be experienced in supporting internet servers running on AIX or Solaris operating systems. All users should have installation and administration skills in the following areas:

- Sun Solaris or AIX
- DB2 or Oracle
- General networking
- Firewalls

Naming conventions

WebSphere Everyplace Server comprises many components and supporting components. The following list gives the full product name of all the components that make up WebSphere Everyplace Server along with any short names used in this book.

Component name	Shortnames
DB2 Universal Database	DB2 Universal Database, DB2
Everyplace Active Session Table	Active Session Table, AST
Everyplace Cookie Proxy	Cookie Proxy
Everyplace Intelligent Notification Services	Intelligent Notification, INS
Everyplace Location Based Services	Location Based Services, LBS
Everyplace Setup Manager	Everyplace Setup Manager, Setup Manager
Everyplace Suite Manager	Everyplace Suite Manager, Suite Manager
Everyplace Synchronization Manager	Synchronization Manager
Everyplace Wireless Gateway	Wireless Gateway
IBM HTTP Server	IBM HTTP Server, IHS
MQSeries Everyplace for Multiplatforms	MQSeries Everyplace
SecureWay Directory	SecureWay Directory, LDAP
Tivoli Personalized Services Manager	Tivoli Personalized Services Manager
Voice Services	Voice Services
WebSEAL-Lite	WebSEAL-Lite
WebSphere Application Server	Application Server
WebSphere Edge Server Caching Proxy	Edge Server Caching Proxy (Web Traffic Express), Caching Proxy
WebSphere Edge Server Load Balancer	Edge Server Load Balancer (Network Dispatcher), Load Balancer
WebSphere Everyplace Server, Service Provider Offering for Multiplatforms	WebSphere Everyplace Server, Everyplace Server
WebSphere Transcoding Publisher	Transcoding Publisher

Related information

- Everyplace Server components
- External information

WebSphere Everyplace Server version 2.1 components

- WebSphere Everyplace Server component overview
- Featured WebSphere Everyplace Server components
- Supporting WebSphere Everyplace Server components
- Third-party components
- Related information

WebSphere Everyplace Server component overview

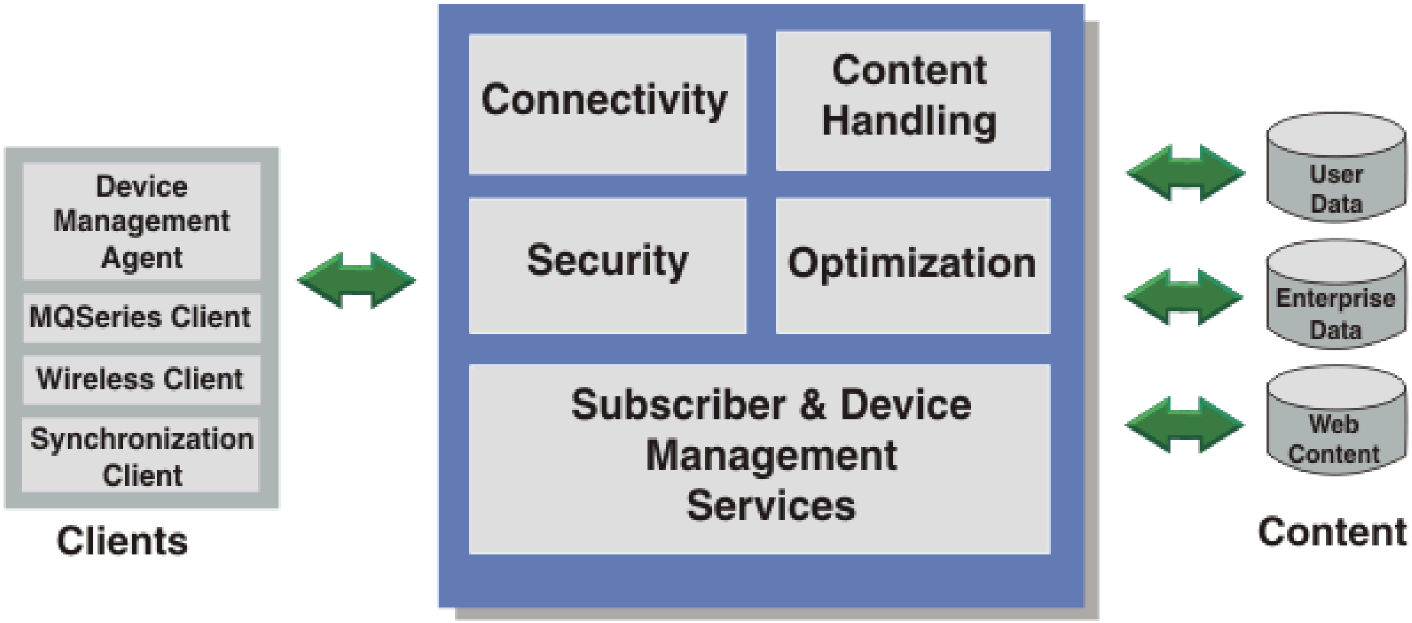
This section provides a detailed description of the Everyplace Server components for Everyplace Server version 2.1.

You only need to install the components that best provide or extend the services you require. More information about the Everyplace Server components and subcomponents is included below.

The Everyplace Server domain consists of a group of servers running Everyplace Server components that are under central administrative control and are within the same protection space (protected area within the same domain name).

The image below shows Everyplace Server providing connectivity between client software on the pervasive devices and internet applications and content.

WebSphere Everyplace Server connecting client devices and Internet data



There are three types of Everyplace Server components:

- Featured Components: the primary Everyplace Server components that are included in the Everyplace Server packaging. These components are installed in the Everyplace Server domain, and some of these components have corresponding subcomponents.
- Supporting Components: components that provide underlying support to the primary Everyplace Server services. These components include IBM HTTP Server and SecureWay Directory
- Third-party Components: components comprised of third-party software packages that are required to support certain Everyplace Server featured components. These components are not included in the Everyplace Server packaging and must be purchased separately.

All of the components, and most of the supporting components, are provided on the Everyplace Server product CDs. Other components need to be downloaded separately and installed prior to installing Everyplace Server.

Featured WebSphere Everyplace Server components

The following products are the featured components of Everyplace Server:

- Everyplace Active Session Table
- Everyplace Cookie Proxy
- Everyplace Intelligent Notification Services
- Everyplace Location Based Services
- Everyplace Suite Manager
- Everyplace Synchronization Manager
- Everyplace Wireless Gateway
- MQSeries Everyplace
- Tivoli Personalized Services Manager
- Voice Services
- WebSEAL-Lite
- WebSphere Edge Server
  - WebSphere Edge Server Caching Proxy
  - WebSphere Edge Server Load Balancer
- WebSphere Transcoding Publisher

Everyplace Active Session Table version 2.1

The Active Session Table Server provides a high speed cache for information about users who are currently connected to the Everyplace Server domain. This replaces the functionality previously provided by the Tivoli Internet Services Manager (TISM) product. Active Session Table Server provides a high speed specialized cache for information about users who are currently connected to the WebSphere Everyplace Server system:

- The Active Session Table client communicates with the Active Session Table server using one or more standard TCP/IP connections. The Active Session Table server continues to provide service for each connection as long as the client remains logged on.

- The IP address of each client that attempts to connect to the Active Session Table server must appear in the accept list of the Active Session Table server. The accept list is one of the Active Session Table server's configuration parameters. The Active Session Table server refuses connections from clients who are not in the accept list.
- The client can send one or more requests on a single connection and the Active Session Table server responds to each request in the order in which they are received.
- The Active Session Table server performs field validation on each request and, if an error is detected, returns an Active Session Table message describing the first error detected.

Everyplace Cookie Proxy version 1.2

Everyplace Server includes Everyplace Cookie Proxy for WebSphere Portal Server version 1.2. Everyplace Cookie Proxy is only available when WebSphere Everyplace Server is installed on Japanese locale machines. This proxy enables users to use i-mode phones when accessing portals built with the WebSphere Portal Server. Messages and other information displayed in the interface are in Japanese only (English is not supported).

Note: Installation of Everyplace Cookie Proxy is only available when installing on machines with a Japanese locale.

Everyplace Intelligent Notification Services version 2.1

Everyplace Intelligent Notification Services delivers messages to pervasive users based on the users' preferences and subscriptions. For example, a user can tell Everyplace Intelligent Notification Server to send them the URL of any Web-based news article published with "pervasive computing" in the headline. The user can also specify message-sending behaviors based on the urgency of the message. For example, if the message is marked FYI, send it to email. If the message is marked urgent, send it to Sametime instant message.

Examples of content sources:

- news
- weather
- stock quotes

Examples of content filters:

- Notify when an article with "pervasive computing" in the title is published.
- Notify when it's going to rain.
- Notify when YourCo stock hits 150.

Examples of message-sending behaviors:

- Send URL to cell phone.
- Send weather info to Sametime.
- Send YourCo stock info to all devices.

Supported Target devices

- Sametime instant messaging
- WAP-enabled phone
- SMS devices (voice)

- Email

Everyplace Intelligent Notification Services consists of the following components:

- **iQueue Server:** Provides trigger management and trigger persistence
- **Universal Notification Dispatcher:** Sends messages to the user by various means, including Instant Messaging, WAP, phone, and e-mail.

For more information about Everyplace Intelligent Notification Services, refer to the Everyplace Intelligent Notification Services documentation included in this InfoCenter. The Intelligent Notification documentation is one of the sets of documentation listed to the left in the navigation frame.

Everyplace Location Based Services version 2.1

WebSphere Everyplace Server Version 2.1 Location Based Services provides user location information to applications. The applications then deliver content based on the user's location information. For example, an application can provide a user with a list of hotels in the area where he/she is currently traveling. Before location information can be used by an application, the administrator must enable the application as a location based application and the user must allow the application to use their location information. The administrator can also enable a user for an application to use their location information.

Location Based Services is shipped with WebSphere Everyplace Server and is a subcomponent of the WebSphere Edge Server Caching Proxy. Location Based Services uses Signal Soft Local.info Server Version 3.2 to provide user location information and Tivoli SecureWay Policy Director to manage privacy settings. Both subcomponents must be installed separately and configured for Location Based Services to work correctly.

Everyplace Suite Manager version 2.1

Everyplace Suite Manager provides a centralized method for launching the administration consoles of the installed WebSphere Everyplace Server components. In addition, Suite Manager obtains information regarding the installed components and the servers where they are installed. From the Suite Manager console, you can make changes to configuration data that is stored in SecureWay Directory (LDAP).

Everyplace Suite Manager allows users to perform the following management tasks from one centralized location:

- Monitor status of Everyplace Server components
- Start and stop Everyplace Server components
- Change initial Everyplace Server configuration settings after Everyplace Server installation is complete
- Launch the administration consoles of individual Everyplace Server components
- View Everyplace Server component logs

Everyplace Synchronization Manager version 1.1.3

Everyplace Synchronization Manager enables handheld computing devices to link remotely to desktop applications. Mobile users can easily synchronize data with Microsoft Exchange, Lotus Notes, DB2 or any ODBC compliant database, such as Oracle or Sybase. The mobile device can synchronize using modem, cellular phone, Internet, Wireless, Intranet, local area network (LAN) or wide area network (WAN). Mobile users can be authenticated through existing Microsoft Exchange or Lotus Notes user data or through a list of users held internally in Everyplace Synchronization Manager. Data can be encrypted for secure transmission. Mobile devices can be automatically backed up or restored and applications can be remotely installed on these devices. Everyplace Synchronization Manager contains the following subcomponents:

- **Everyplace Synchronization Manager Service**: Handles the request from the mobile device, manages security, and performs all the data transfers between the mobile and the enterprise data sources. Runs on a Unix server.
- **Everyplace Synchronization Manager Admin**: Enables the administrator to set up or modify the synchronization performed by the Synchronization Manager service. Uses wizards or intuitive forms. Runs on a UNIX server.
- **Exchange Connector**: Enables Synchronization Manager to synchronize with Microsoft Exchange Server. Runs on Windows NT 4.0 or 2000.
- **Notes Connector**: Enables Synchronization Manager to synchronize with Lotus Notes. Runs on a UNIX server.
- **Everyplace Synchronization Proxy**: Mobile devices may synchronize either directly (through dial-up or packet network) to the Synchronization Manager Service or indirectly with a serial cable to a desktop PC which then connects to the Synchronization Manager Service. Everyplace Synchronization Proxy must be installed and running on the desktop PC to synchronize through cable. Runs on Windows.
- **Everyplace Synchronization Client**: Enables the mobile device to synchronize with enterprise data sources through the Synchronization Manager Service. Clients are packaged with the Windows install library.

Note: This component is only available with specific Everyplace Server license keys.

IBM Everyplace Wireless Gateway for Mulitiplatforms version 2.1

IBM Everyplace Wireless Gateway provides a communications platform that enables Internet Protocol and Wireless Access Protocol (WAP) applications to run in a wireless and wired environment. Wireless Gateway provides mobile devices containing the Wireless Client with access to host and network resources through radio and dial-up networks. It can encrypt, compress,and minimize the data that passes through the wireless link, thereby increasing the speed of messaging. The Wireless Gateway contains the following subcomponents:

- **Wireless Gatekeeper**: A Java-based administration tool for the Wireless Gateway and wireless resources. It enables an administrator to configure wireless and wireless access protocol (WAP) gateways, add users and mobile devices, define and group wireless resources, and assign administrators to wireless resources.
- **Ardis Support**: Enables use of the advanced radio data information services protocol. Motient is the network provider.
- **Dataradio Support**: Enables use of the Dataradio network provider.
- **DataTAC Support**: Enables use of the DataTAC 5000 and DataTAC 6000 networks.
- **Dial Support**: Enables use of dial-capable digital and analog networks such as global system for mobile communication (GSM), advanced mobile phone service (AMPS), public switched telephone network (PSTN), and integrated service digital network (ISDN) networks. Native point-to-point protocol (PPP) is also supported over these networks.

- **IP LAN Support**: Enables use of a LAN-based network provider and all IP-based mobile devices, such as cellular digital packet data (CDPD), and general packet radio service (GPRS), among others. IP LAN Support works for wired environments and for any two nodes on a network. Using the two network nodes, you can create a secure tunnel, which functions as a virtual private network (VPN), between the nodes.
- **Mobitex Support**: Enables use of networks that contain the Mobitex protocol. These networks include BellSouth Wireless, CanTel, and Norcom Satellite.
- **Modacom Support**: Enables use of the Modacom network provider.
- **Motorola PMR**: Enables communication with one or more RNC-3000 network controllers in a Motorola private mobile radio (PMR) network.
- **Short Message Service (SMS)**: Enables support of the UCP/EMI and SMPP SMS protocols. Commonly used for short messages in GSM-SMS network.
- **SMTP:** Enables support of the SMTP e-mail messaging and allows the Everyplace Wireless Gateway's messaging gateway to connect to an SMTP mail server and forward short messages to end user that can be addressed with an e-mail address.
- **Simple Network Paging Protocol (SNPP):** Enables support of the Simple Network Paging Protocol which is used to send messages to Short Message Service Centers that support the SNPP protocol.
- **Wireless Client**: This optional interface starts and stops communication with a Wireless Gateway. Wireless Client shields network-specific details inside the interface layer and allows IP applications on a mobile computer to run over a wireless network. For example, a radio network would not require any specialized communication protocols for use by a mobile device. Wireless Client is not installed on an Everyplace Server machine, but rather on the client device.

Note: IBM Everyplace Wireless Gateway is only available with specific WebSphere Everyplace Server license keys. See Install Strategies for information on license keys.

MQSeries Everyplace for Multiplatforms version 1.2

MQSeries Everyplace provides assured messaging capability between devices and any MQSeries family platform. It extends secure messaging to include dependable communications with mobile workers. It connects laptops, servers, PDAs, phones, and unattended devices, such as sensors, to MQSeries networks. This enables users to perform business functions, including e-mail access, stock purchase, or order placement through their mobile devices. MQSeries Everyplace consists of Java(R) and C components enabling solution developers to create an MQSeries Everyplace gateway and client on a variety of devices and platforms.

The native C client version of MQSeries Everyplace is not installed with Everyplace Server. This version can be downloaded from:

http://www.ibm.com/software/ts/mqseries/

Tivoli Personalized Services Manager version 1.2

Tivoli Personalized Services Manager enables service providers to centrally manage subscribers and devices. Management includes enrolling subscribers and devices, providing self care and customer care, maintaining and billing subscriber accounts, and submitting jobs such as software distribution to devices, among others. Tivoli Personalized Services Manager contains the following subcomponents:

- **Device Manager**: Helps service providers manage their subscribers' pervasive devices, including PDAs, subnotebooks, and other devices. Device Manager can identify, configure, and distribute software to any device that the Device Manager and the service provider support.

- **Enrollment Server**: Provides a subscriber and device enrollment engine for an ISP, including a customizable set of screens with unique banners, messages, billing plans, and payment options.
- **Database Integration**: Enables the installation program or the user to create either a DB2 or Oracle database. If you install Tivoli Personalized Services Manager, you must install this subcomponent.
- **Customer Care**: Enables representatives to open new or child accounts and deactivate or reactivate accounts. It also enables representatives to view and update personal information, service plans, payment methods, and e-mail settings.
- **Self Care**: Enables subscribers to modify some of their profile data, such as address, telephone data and payment method. It also allows subscribers to select a new service plan.
- **System Management**: Provides the ability to set up groups of subscribers, domains, and membership plans and deals. It also enables the administrator to access subscriber profiles.
- **Everyplace Server Enabler**: Allows Tivoli Personalized Services Manager to manage its subscriber database in SecureWay Directory.
- **Portal Toolkit**: Provides portlets in the Everyplace Server environment and enhances personalization by allowing the development of portal pages. It can also delegate authentication to the Everyplace Server Authentication Proxy. The web authentication server interfaces with the Everyplace Server Authentication Proxy and when used, users are authenticated and device type identification is achieved.

Voice Services version 2.1

Voice Services allows users to authenticate their identity to the Everyplace Server domain and gain access to Everyplace Server functions using a telephone. Voice Services is a voice version of the authentication process for web browser authentication.

Voice Services is compliant with VXML 1.0 Voice Services. However, we have only tested and verified compliance with the AIX server and Motorola Server.

To run Voice Services, there are three hardware/software components which must be installed and set up correctly prior to setting up Voice Services.

- **Via Voice -** Via Voice delivers low cost, very low resource Speech Recognition. Via Voice provides command and control, RTOS Support, speaker dependent language independent, speaker independent language independent, and a 8 kHz sampling rate. In Voice Services, Via Voice translates the users speech when Voice Services is authenticating the user. The text is then used by the VXML file to determine the user is authorized.
- **Direct Talk -** Direct Talk allows enterprises to deploy voice-enabled applications on a traditional IVR network infrastructure. Direct Talk uses the existing DT platform and IVR network infrastructure, enables current Direct Talk users to transition to a Web-development paradigm using Voice XML, and supports the Voice XML 1.0 specification. In Voice Services, Direct Talk translates the VXML file from text to speech. This allows the user to here the speech version of the VXML file.
- **Voice Server -** Voice Server allows you to deploy web-based Voice applications written in VXML. In Voice Services, the Voice Server is the server where the

WebSphere Everyplace Server Voice Services application and VXML file is installed.

Note: The Voice Server only runs on Windows NT.

WebSEAL-Lite version 2.1

WebSEAL-Lite is the central point of user authentication for the Everyplace Server domain. It authenticates users defined to the Everyplace Server domain when they attempt to access Everyplace Server services. WebSEAL-Lite also allows you to use gateways other than Everyplace Wireless Gateway if desired. At least one version of WebSEAL-Lite is required in the Everyplace Server domain to enable integration of most Everyplace Server components. It is the point of entry to the Everyplace Server domain for devices that do not connect through Everyplace Wireless Gateway and is the next non-firewall hop for connections through Everyplace Wireless Gateway.

WebSEAL-Lite runs as a plug-in to the Edge Server Caching Proxy. The Caching Proxy is a prerequisite for WebSEAL-Lite and must be installed on the same machine as WebSEAL-Lite. WebSEAL-Lite can be configured in one of two modes:

- **Authentication proxy**: Performs user authentication based on HTTP Authenticate headers. In an Everyplace Server domain where the authentication proxy is installed, no other origin server (content or application server) in the Everyplace Server domain may do its own user authentication. Users authenticated through the authentication proxy may not access content outside of the Everyplace Server domain.
- **Transparent authentication proxy**: Performs user authentication based on HTTP Proxy-Authenticate headers. In an Everyplace Server domain where transparent proxy is installed, origin servers (content or application servers) in the Everyplace Server domain may do their own user authentication. The transparent authentication proxy allows users to access material outside the Everyplace Server domain.

Note: WebSEAL-Lite allows for single user sign-on (user ID and password) for all services within the Everyplace Server domain. With this feature, user authentication only needs to be done once to access services requiring a user ID and password. Authentication is still needed for services outside the Everyplace Server domain.

For example: Users log on to an enterprise site that uses Everyplace Server and give their user ID and password, which is then authenticated by WebSEAL-Lite. If users want to change their password (performed by Tivoli Personalized Services Manager), they do not have to enter a user ID and password again to access this service.

WebSphere Transcoding Publisher version 3.5.1

The Transcoding Publisher adapts Web content based on destination device characteristics and network service level. You can enhance the performance of the Transcoding Publisher by also installing Edge Server Caching Proxy, which stores transcoded material. This removes the necessity of retranscoding Web pages each time they are retrieved.

Note: WebSphere Transcoding Publisher is intended to be deployed as a proxy in the Everyplace Server domain. It is not intended to be used as a servlet or a JavaBean within the Everyplace Server domain.

Websphere Edge Server version 1.03

WebSphere Edge Server Caching Proxy version 3.6.01

The Caching Proxy, as known as Web Traffic Express, retrieves Internet data for multiple browser clients. It also acts as a caching server and content filter, reducing the time needed to retrieve information from the Internet and filtering Internet data for multiple browser clients.

WebSphere Edge Server Load Balancer version 3.6

The Load Balancer, also known as Network Dispatcher, provides dynamic load balancing, scalability, and high availability for servers, boosting overall server performance by automatically finding the optimal server within a group of servers to handle each incoming request. It can be used with Web servers, e-mail servers, distributed parallel database queries, and other Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) applications. The Load Balancer contains the following subcomponents:

- **Content Based Routing**: Performs balancing in one of two ways:
  - For HTTP, Content Based Routing performs balancing based on the content of an HTTP client request. This method requires the Caching Proxy on the same machine.
  - For IMAP and POP3, Content Based Routing performs balancing on IMAP or POP3 mail servers. It selects the appropriate server based on the user ID and password provided by the client and does not require the Caching Proxy.
- **Dispatcher**: An IP packet-level load balancer. It provides high performance, low latency load balancing using weights and measurements that are dynamically set. It also provides built-in support for protocols such as HTTP, FTP, SSL, NNTP, IMAP, POP3, SMTP, and Telnet, but can be extended to support both TCP and UDP.
- **Interactive Session Support**: Balances the load on servers using a domain name server. This is done by communicating with server agents that are used to monitor the load and then altering the IP address returned to the client based on this load. Interactive Session Support can also provide the same server load information to the Dispatcher subcomponent.

Supporting WebSphere Everyplace Server components

The following products are considered Everyplace Server support components.

- Included on the Everyplace Server CDs and installed by Everyplace Server:
  - DB2 Universal Database
  - IBM HTTP Server

- Java Runtime Environment
- SecureWay Directory
- WebSphere Application Server Advanced Edition

- Included on the Everyplace Server CDs but not installed by Everyplace Server Setup Manager:
  - Tivoli SecureWay Policy Director
  - Sametime Everyplace

Included on the Everyplace Server CDs and installed by Everyplace Server Setup Manager:

DB2 Universal Database version 7.1, Fixpack 2a

DB2 Universal Database is a Web enabled relational database management system, supporting many levels of complexity in database environments.

IBM HTTP Server version 1.3.12.3

An IBM enhanced Web server based on the Apache Web server. IBM HTTP Server supports both the secure sockets layer (SSL) version 2 and SSL version 3 protocols for secure connections. It also includes a cache accelerator for improved performance when serving static Web pages.

Java Runtime Environment version 1.3

Contains the software and tools used to compile, debug, and run applets and applications written using the Java programming language.

SecureWay Directory version 3.2.1

A Lightweight Directory Access Protocol (LDAP) directory that runs as a stand-alone daemon. It is based on a client/server model that provides client access to an LDAP server. SecureWay Directory provides an easy way to maintain directory information in a central location for storage, updating, retrieval, and exchange.

WebSphere Application Server Advanced Edition version 3.5.4

Enables Web transactions and interactions with a robust deployment environment for e-business applications. It provides a portable, Java-based Web application deployment platform focused on supporting and executing servlets, JavaBeans, JavaServer Pages (JSP) files, and enterprise beans.

Included on the Everyplace Server CDs but not installed by Everyplace Server Setup Manager:

Tivoli SecureWay Policy Director version 3.7.1

Tivoli SecureWay Policy Director is a robust and secure policy management tool for e-business and distributed applications. It uniquely addresses the challenges of e-business security-escalating costs, growing complexity, and the inability to implement security policies across platforms.

Tivoli SecureWay Policy Director is designed to unite core security technologies around common security policies. This helps reduce implementation time and management complexity, thereby lowering the total cost of secure computing.

Product features:

- Provides access control to Web objects
- Adds centralized security to your existing Web and TCP/IP applications
- Enables replication and load balancing
- Provides a consistent, manageable access control policy
- Offers extensible authentication and authorization
- Delivers secure remote access and personalized access
- Offers one-time authentication capability with access to multiple Web resources
- Reduces administration costs
- Supports Public Key Infrastructure (PKI)

Note: Tivoli SecureWay Policy Director is not installed using the Everyplace Server installation program. It must be installed separately. See the documentation on the Tivoli SecureWay Policy Director CD for installation instructions.

Sametime Everyplace version 1.0

Sametime Everyplace extends the capabilities of Sametime to WAP-enabled devices such as mobile phones. It allows you to chat with other Sametime users from your mobile phone, whether they are using mobile devices, or whether they are using Sametime Connect from their desktop. Sametime Everyplace allows you to:

- View status information to see whether a person is a mobile user and to see who is online
- Create Contact lists and search for users
- Use the Chat function for live instant messaging over WAP
- Invite multiple Sametime users to join a group chat. Send notifications, via Short Message Service (SMS), to mobile users who are logged on to Sametime, but not currently using Sametime
- Send e-mail to users who are not online

Note: Sametime Everyplace is not installed using the Everyplace Server installation program. It must be installed separately. See the documentation on the Sametime Everyplace CD for installation instructions.

Third-party components

The following components are not packaged with Everyplace Server; however, they are required to support certain Everyplace Server components:

- Netscape Navigator or Netscape Communicator
- SignalSoft

Netscape Navigator version 4.08 (or higher) or Netscape Communicator version 4.5 (or higher)

Displays Internet Web pages and other HTML-based documents. See the Web site *http://www.netscape.com/computing/download/index.html* for downloading and installation instructions.

SignalSoft

SignalSoft is a third-party software package that is required for use with Location Based Services. You can purchase SignalSoft from the following location:

www.signalsoftcorp.com

Related information

- Requirements and prerequisites

External information

- Redbooks
- Development tools
- Related information

Redbooks

You can access the following Redbooks for Everyplace Server and Everyplace Server components through the IBM Redbook site:

- Extending e-business to Pervasive Computing Devices Using IBM WebSphere Everyplace Suite version 1.1.2, SG24-5996-00
- An Introduction to IBM WebSphere Everyplace Suite version 1.1 Accessing Web and Enterprise Applications, SG24-5995-00
- Using IBM WebSphere Everyplace Suite version1.1.2, SG24-5996-00
- IBM WebSphere Transcoding Publisher V1.1: Extending Web Applications to the Pervasive World, SG24-5965-00
- Mobile Computing: The eNetwork Wireless Solution, SG24-5299-00
- Web Caching and Filtering with IBM WebSphere Performance Pack, REDP0009 (redpaper)
- WebSphere V3 Performance Tuning Guide, SG24-5657-00
- IBM WebSphere Performance Pack: Caching and Filtering with IBM Web Traffic Express, SG24-5859-00
- IBM WebSphere Performance Pack: Load Balancing with IBM SecureWay Network Dispatcher, SG24-5858-00
- Using LDAP for Directory Integration A look at IBM SecureWay Directory, Active Directory and Domino, SG24-6163-00

Development tools

The WebSphere Everyplace Server System Development Kit (SDK) is available to help you build wireless applications that run on various Everyplace Server components. The WebSphere Everyplace Server SDK, available through the IBM PartnerWorld(R) program, allows you to create and test applications, and includes WAP Client-Proxy, phone and gateway simulators, WML, Push, VoiceXML samples, and documentation. For more information, see http://www.ibm.com/pvc/tech/wes_sdk.shtml.

Related information

- Everyplace Server components

Extend WebSphere Everyplace Server functionality

The integration of several additional tools with WebSphere Everyplace Server can further extend the functionality of your enterprise. These features may not be included with Everyplace Server, but they can be used in conjunction with Everyplace Server components to further increase its capabilities. This section details the benefits of several available features.

- WebSphere Portal Server with Everyplace Server
- Domino with Everyplace Server
- Instant messaging with Sametime
- Related information

WebSphere Portal Server with Everyplace Server

Everyplace Server works hand in hand with WebSphere Portal Server version 1.2 to create a dynamic environment for your customers. Enterprises that add Portal Server to the Everyplace Server have the ability to offer their users wireless portals. A portal provides a single point of interaction with diverse information, business processes, and people, all personalized to an end user's needs and responsibilties. Implementing both Everyplace Server and Portal Server allows end users to create and access their customized portal from any location using assorted wireless and wireline devices while maintaining the security expected from Everyplace Server.

One feature that greatly extends the reach of Everyplace Server is Wireless Access Protocol (WAP) email. The wireless capabilities that Everyplace Server provides combined with Portal Server's email portlet gives end users wireless access to their POP3 and IMAP email accounts. The integration of Everyplace Server with Portal Server provides users with the ability to check their email accounts through a WAP device.

Both Everyplace Server and Portal Server also support single sign-on, which allows for a more versatile, user friendly experience while maintaining a secure environment.

For more information on how WebSphere Portal Server can enhance your Everyplace Server domain, go to:
http://www.ibm.com/software/webservers/portal/

Domino with Everyplace Server

Enterprises that run Domino with Everyplace Server benefit from the extended functionality of the two products. Together, they provide the tools necessary to rapidly design, test, revise, and deploy applications to meet today's changing business needs. The Everyplace Server-Domino combination, unlike other solutions, delivers fully functional, high performance support for distributed transactions while easily making use of the other's services.

Single sign-on is just one of the features offered by Everyplace Server and Domino. When users sign on to Everyplace Server, they are automatically signed in to Domino without taking any further action on their part. You must configure both Everyplace Server and Domino to accept single sign-on to take advantage of this feature. You must also configure Domino to accept LTPA tokens for single sign-on capabilities. Required Everyplace Server components include Edge Server Caching Proxy and WebSEAL-Lite. Everyplace Server includes Domino Application Server and Domino Everyplace, which consists of Domino Access and SMS.

Security Note: WebSEAL-Lite supports the creation of LTPA tokens and sending of LTPA cookies, required for Domino single sign-on, to end users.

For more information on Domino, go to:
http://www.lotus.com/home.nsf/welcome/domino

Instant messaging with Sametime

Sametime delivers powerful real-time collaboration for businesses. Users can instant message people across the hall or around the world. Sametime also interoperates with AOL Instant Messaging and Microsoft Netmeeting, so business meetings can take place across multiple programs. With an infrastructure that IT managers can understand and trust, Sametime offers enterprise-wide security, scalabitilty, and the management control they expect. Sametime greatly enhances real-time business communication and increases productivity.

Sametime has now broadened its reach to include wireless devices with Sametime Everyplace. The WebSphere Everyplace Server includes Sametime Everyplace, allowing you to extend all the real-time functionality of Sametime to mobile devices. Everyplace Server includes Sametime Everyplace as an additional feature. With Everyplace Server and Sametime Everyplace, mobile users are able to create and modify Connect lists, chat synchronously with other Sametime users, and integrate phone features, such as mobile number access and instant call requests, from within the Sametime environment.

Note: An existing Sametime deployment is required and not included with Everyplace Server.

For more information on Sametime, go to:

http://www.lotus.com/home.nsf/welcome/sametime

Related information

- WebSEAL-Lite
- Sametime Everyplace

Supported pervasive devices and network types

- Device table
- Related information

Supported device and network types

The Everyplace Server components support the devices and network types listed below. Except where noted, Everyplace Server includes wireless client code on the specified platforms for full security, connectivity, and optimization functions.

Everyplace Server Component	Supported Platforms and Network Types
Everyplace Synchronization Manager	● Palm OS ● Windows CE ● Pocket PC ● EPOC (supported at a future time)
Everyplace Wireless Gateway	● Microsoft(R) Windows(R) CE 2.0 and 2.11 ● Windows CE (PPC)[1] ● Windows CE V3.0[1] ● Microsoft Windows 95 and 98 ● Microsoft Windows NT ● Microsoft Windows 2000 ● Palm OS ● Pre-EPOC WAP phones ● QNX/Neutrino ● WAP phones (1.1 and 1.2)[2]

MQSeries Everyplace	● Windows CE ● Microsoft Windows 95 ● Microsoft Windows 98 ● Microsoft Windows NT ● Microsoft Windows 2000 ● Palm OS ● EPOC ● Any device running Java JVM 1.1 or later
Tivoli Personalized Services Manager (Device Manager)	● Windows CE ● Palm OS ● QNX/Neutrino ● NetVista Internet Appliance
WebSphere Transcoding Publisher	● Windows CE ● Microsoft Windows 95 ● Microsoft Windows 98 ● Microsoft Windows NT ● Microsoft Windows 2000 ● Palm OS ● EPOC ● Pre-EPOC WAP phones ● i-mode

Notes:

1. These devices are supported as WAP-capable clients (with no IBM code needed on the device) or as a standard PPP-capable client. Full connectivity is supported but with somewhat less optimization and security and no data encryption.
2. These devices are supported as WAP-capable clients only (with no IBM code needed on the device).

Related information

- Everyplace Server components

Language Support

This section contains information on installing WebSphere Everyplace Server in the following language environments:

Japanese-specific information

Everyplace Server supports the following language environments for Japanese:

- AIX: Primary Language Environment: Japanese PC (`Ja_JP`)
- Solaris: Default Language: Japanes EUC (`ja`)

If you install Everyplace Wireless Gateway on a Japanese AIX system, you must install the Japanese EUC (`ja_JP`) system locale.

If you install Tivoli Personalized Services Manager on a Japanese system, you must install the Japanese UTF-8 (`JA_JP`) system locale for AIX or the Japanese UTF-8 (`ja_JP.UTF-8`) system locale for Solaris.

Using the Tivoli Personalized Services Manager Self Care subcomponent in a Japanese environment

To use the Tivoli Personalized Services Manager Self Care subcomponent in a Japanese environment, you must perform the following:

- On WebSEAL-Lite, edit the following file:

  `/opt/IBMWTE/usr/internet/etc/ibmproxy.conf`

  Refer to "Configuring WebSEAL-Lite " in the Configuring section.
- Add the following lines to the "SELFCARE" section of the `ibmproxy.conf` file:

  ```
  #####
    Request URL:http://authserver:port/tsm_sc/ja/selfcare.html
    Proxy /tsm_sc/ja/* http://&lt;hostname&gt;.&lt;domain&gt;:15080/ja/*
  ```

CD contents

Getting started

WebSphere Everyplace Server is made up of many CDs. Disc 1 contains the Everyplace Server documentation and the installation program.

Disc 1 is structured as follows:

`/info/readme.htm`

Everyplace Server readme.

`/info/infocenter/index.html`

Everyplace Server InfoCenter.

`/install`

Installation program files. See Related information to learn how to start the installation.

Everyplace Server CD contents

Disc Number	Contents of CD
Disc 1	● Everyplace Server Setup Manager, installation program for WebSphere Everyplace Suite ● Everyplace Suite Manager configuration and administration interface ● Everyplace Server Documentation, including the Everyplace Server InfoCenter and the README ● Java Runtime Environment 1.3 (AIX and Sun Solaris) ● IBM HTTP Server
Disc 2	● DB2 Universal Database for AIX (for Single Byte Character Sets)

Disc 3	● DB2 Universal Database for Sun Solaris (for Single Byte Character Sets)
Disc 4	● MQSeries Everyplace ● SecureWay Directory
Disc 5	● WebSphere Application Server Advanced Edition
Disc 6	● WebSphere Edge Server ○ WebSphere Edge Server, Caching Proxy (Web Traffic Express) ○ WebSphere Edge Server, Load Balancer (Network Dispatcher)
Disc 7	● Everyplace Wireless Gateway (except for the Wireless Client subcomponent)
Disc 8	● WebSphere Transcoding Publisher
Disc 9	● Tivoli Personalized Services Manager
Disc 10	● Everyplace Active Session Table ● Everyplace Cookie Proxy (installed on Japanese locales only)* ● Everyplace Intelligent Notification Services ● Everyplace Location Based Services ● Everyplace Synchronization Manager ● Voice Services ● WebSEAL-Lite
Disc 11	● DB2 Universal Database for AIX (for Double Byte Character Sets)
Disc 12	● DB2 Universal Database for Sun Solaris (for Double Byte Character Sets)
Disc 13	● DB2 Universal Database Fixpack 2a
The following CDs contain Tivoli SecureWay Policy Director. WebSphere Everyplace Server Setup Manager does not install this component. Refer to the documentation in the /Doc directory on these CDs for installation instruction.	
Disc 14	● Tivoli SecureWay Policy Director for AIX
Disc 15	● Tivoli SecureWay Policy Director for Solaris
Disc 16	● Tivoli SecureWay Policy Director Console
Disc 17	● Tivoli SecureWay Policy Director for Windows NT

Sametime Everyplace CDs

Sametime Everyplace is not installed by Setup Manager. Refer to Disc 1 below to start the installation.

Disc Number	Contents of CD
Disc 1	● Lotus Sametime Everyplace Server for Windows NT
Disc 2	● Lotus Domino Everyplace SMS for Windows NT
Disc 3	● Lotus Domino Application Server for Windows NT

Building the installation image

If you already have the Everyplace Server installation CDs you can ignore this section. Everyplace Server is made up of many large files that are downloadable from the Internet. Make sure there is sufficient space to download the CD images and approximately 700MB of space per CD image to expand the contents. Each file corresponds to a CD image used to install Everyplace Server. The files are in a compressed tar format (with a `.taz` extension) and must be extracted.

To build the install image:

1. Create the directory for each CD image to be extracted into by entering:

   ```
   mkdir /cd1

   mkdir /cd2

   mkdir /cd3
   ```

   and so on for all the CD images.
2. Download the CD images into the corresponding directories.
3. Uncompress each compressed CD image by entering the command:

   ```
   gzip -dvf cdXimage.taz
   ```

   Where `cdXimage.taz` is the name of the downloaded CD image and X is the Disc number corresponding to each of the CDs. Be sure to maintain the numeric order of the file names from the compressed files to the CD install images, creating ordered file names such as: `cd1,cd2,cd3...cd10`.
4. Extract the files from each CD image. For example, for Disc 1, enter:

   ```
   tar -hxvf cd1image.tar
   ```
5. At this point you can create the CDs that will be used for installation or install directly from the extracted install images.

Related information

- Prepare for installation
- Installation steps

Installation Strategies

This section describes installation fundamentals for WebSphere Everyplace Server. It includes implementation scenarios and deployment models based on potential product keys. Implementation scenarios illustrate how Everyplace Server components function and how communications might flow within the Everyplace Sever domain. Each deployment model suggests the installation components necessary for the described scenario.

- Overview
- License key options
- WebSphere Everyplace Server, Service Provider Offering
- WebSphere Everyplace Server, Service Provider Base Offering
- Everyplace Wireless Gateway for WebSphere Everyplace Server, Service Provider Base Offering
- Everyplace Synchronization Manager for WebSphere Everyplace Server, Service Provider Base Offering
- Related Information

Overview

The installation of Everyplace Server is unique for most implementations. The Everyplace Server domain consists of a group of servers in an enterprise that are under central administrative control and are within the same protection space. Typically, there are a number of servers within a local area network (LAN) that have one or more Everyplace Server components installed on them. For example, there may be a cluster of four servers running WebSphere Transcoding Publisher and a cluster of eight servers running Everyplace Wireless Gateway.

Everyplace Server requires the use of a Lightweight Directory Access Protocol (LDAP) for information sharing and cross-component communications. Everyplace Server components use the SecureWay Directory as a common information platform to maintain a seamless integration within the Everyplace Server domain. Everyplace Server relies on a specific directory schema that is only implemented in SecureWay Directory version 3.2. Therefore, SecureWay Directory can be seen as a prerequisite for all the Everyplace Server components. It is strongly recommended that SecureWay Directory be deployed within any Everyplace Server domain.

Note: Be sure to consult an IBM technical representative before attempting to install or use Everyplace Server with an LDAP implementation other than SecureWay Directory.

License key options

There are many components included in each version of Everyplace Server, Service Provider Offering. Depending on your needs, you may have requested an license key that does not include all Everyplace Server components. Depending on which key you have some components described in this documentation may not be available for installation.

If you have any key other than the full installation key, you can upgrade to enable more components. The installation program, Setup Manager, only requires the highest level key to enable the components you want. For example, if you have the base key and upgrade to get Everyplace

Wireless Gateway, you only need the Everyplace Wireless Gateway key. The table below provides the key names and describes your upgrade options.

License key name	Description	Upgrade options	Key ending
WebSphere Everyplace Server, Service Provider Offering	Full installation	Not applicable, full installation	xxxx-xxxx-xxxx-xxxx-7CUB
WebSphere Everyplace Server, Service Provider Base Offering	Base installation	Upgrade available via special bid only. Consult your IBM representative for details.	xxxx-xxxx-xxxx-xxxx-7MSP
Everyplace Wireless Gateway for WebSphere Everyplace Server, Service Provider Base Offering	Everyplace Wireless Gateway. Requires license key for WebSphere Everyplace Server, Service Provider Base Offering	Upgrade available via special bid only. Consult your IBM representative for details.	xxxx-xxxx-xxxx-xxxx-7IAH
Everyplace Synchronization Manager for WebSphere Everyplace Server, Service Provider Base Offering	Everyplace Synchronization Manager. Requires license key for WebSphere Everyplace Server, Service Provider Base Offering	Upgrade available via special bid only. Consult your IBM representative for details.	xxxx-xxxx-xxxx-xxxx-QJEO

WebSphere Everyplace Server, Service Provider Offering

The Everyplace Server, Service Provider Offering key opens all components in the Everyplace Server, including the Everyplace Wireless Gateway and Everyplace Synchronization Manager. This deployment could be typical for an enterprise that wants a complete solution for pervasive device support along with traditional wired internet services. See "Content provider -- start-up portal company" and "Enterprise customer -- package delivery company" for a description of enterprises that benefit from a complete or limited installation of Everyplace Server with Wireless Gateway and Synchronization Manager.

Implementation Scenario: Content provider -- start-up portal company
A start-up portal company wants a complete solution for device support. This is a company with no existing database of users. The company provides content through an internet portal page that users can customize to their personal tastes and needs. The company also wants to offer services such as e-mail and search engine capability for both wired and wireless devices. This company requires a complete suite of services including transcoding, device management, data synchronization, and assured messaging. The company also needs integrated network access support (NAS) and wireless access protocol (WAP) connectivity. One example of how

communications flow within this Everyplace Server domain could be:

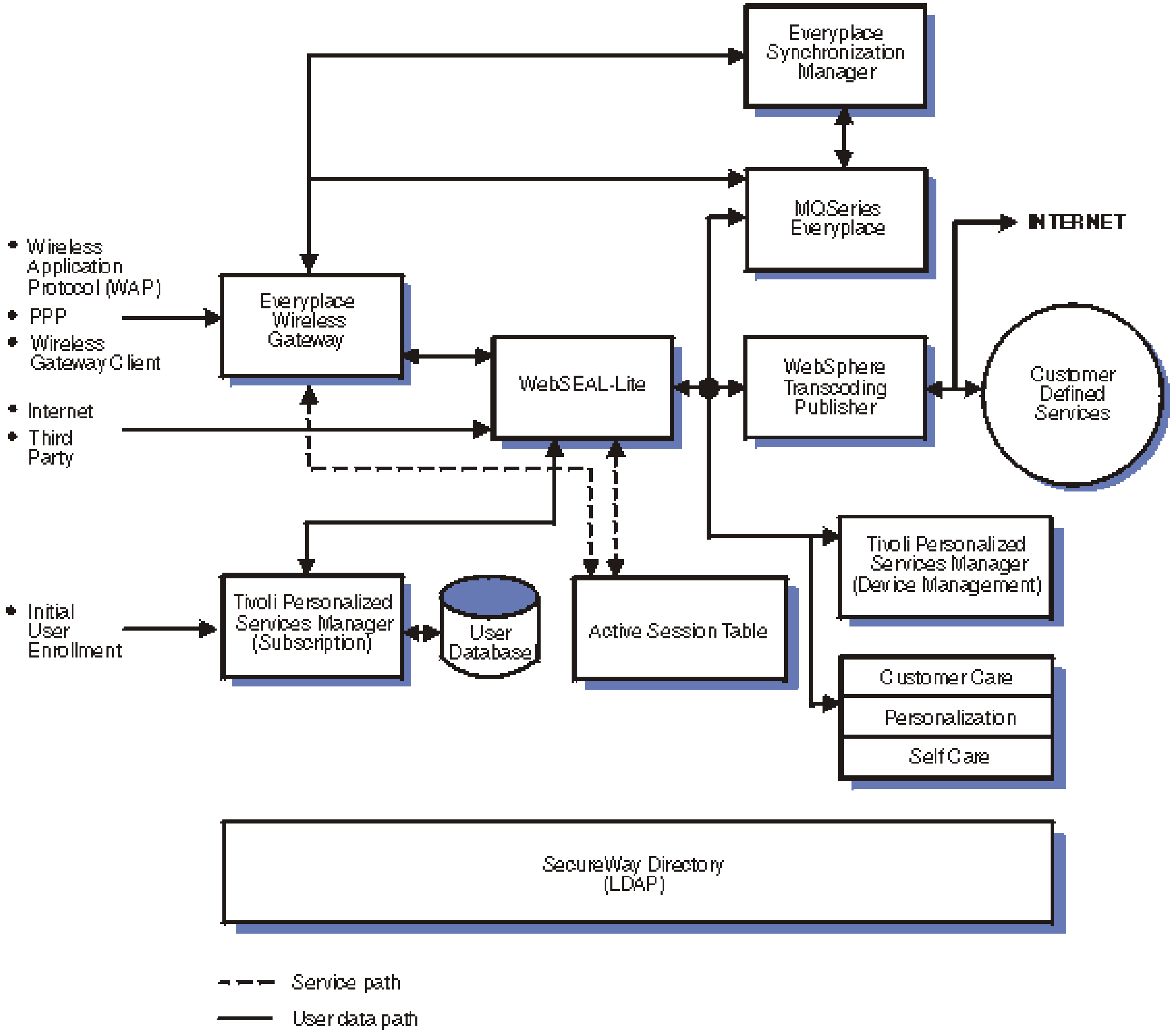
- A user wishes to access personal e-mail using a cellular phone.
- After the connection is made, the Wireless Gateway receives the request.
- The Edge Server Load Balancer dispatches the request to WebSEAL-Lite.
- WebSEAL-Lite performs the authentication which allows the user access.
- The Edge Server Load Balancer dispatches the request to the appropriate WebSphere Transcoding Publisher server.
- The Transcoding Publisher forwards the request to the server handling the back end customer defined services (possibly the WebSphere Application Server).
- The Application Server returns the requested e-mail data to the Transcoding Publisher.
- The Transcoding Publisher reformats the e-mail data according to the cell phone's display specifications and returns it to the Wireless Gateway.
- The Wireless Gateway forwards the reformatted e-mail data to the cell phone user.

Deployment Model

This scenario would require the following Everyplace Server components:

- WebSphere Transcoding Publisher -- data transcoding
- MQSeries Everyplace -- assured messaging
- Everyplace Wireless Gateway -- wireless device connectivity (WAP support)
- Edge Server Caching Proxy -- performance optimization
- Edge Server Load Balancer -- performance optimization
- WebSEAL-Lite -- user authentication and security
- Everyplace Synchronization Manager -- data synchronization
- Tivoli Personalized Services Manager -- subscriber and device management
- IBM DB2 Universal Database -- data storage

Everyplace Server installation allows for flexibility in component installation. The image below represents an Everyplace Server domain where all key components are deployed.


Everyplace Synchronization Manager
MQSeries Everyplace
INTERNET
• Wireless Application Protocol (WAP)
• PPP
• Wireless Gateway Client
Everyplace Wireless Gateway
WebSEAL-Lite
WebSphere Transcoding Publisher
Customer Defined Services
• Internet
• Third Party
Tivoli Personalized Services Manager (Device Management)
• Initial User Enrollment
Tivoli Personalized Services Manager (Subscription)
User Database
Active Session Table
Customer Care
Personalization
Self Care
SecureWay Directory (LDAP)
Service path
User data path

Complete Everyplace Server installation

Implementation Scenario: Enterprise customer -- package delivery company

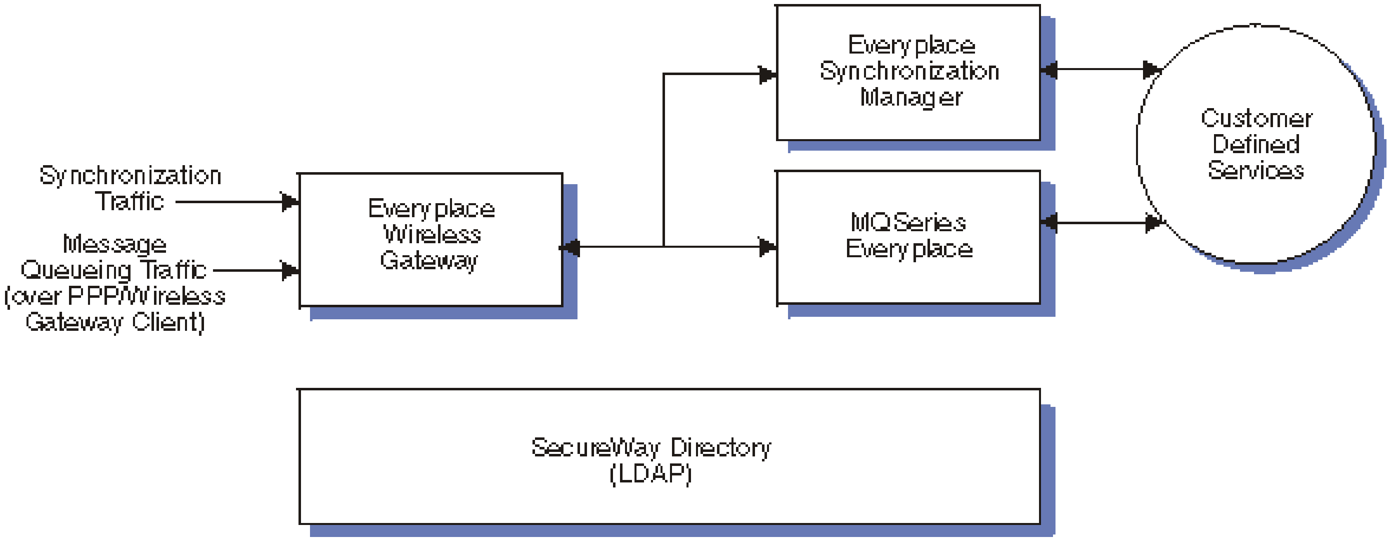
An overnight delivery company wants to equip its delivery personnel with wireless devices to record and track delivery of packages in the field. The company already has a device and user support infrastructure in place, including user and subscriber management, so a limited installation is acceptable. It is not important that network access infrastructure be integrated with the rest of its device infrastructure. Communication flow within this Everyplace Server domain may be:

- A field employee uses a mobile device to connect.
- The Wireless Gateway receives the request.
- The Synchronization Manager updates the database with the employee's entries.
- The Wireless Gateway forwards any new data back to to the employee's mobile device.

Deployment Model
Deployment for this scenario requires the following Everyplace Server components:

- Everyplace Wireless Gateway -- wireless device connectivity
- MQSeries Everyplace -- assured messaging
- Everyplace Synchronization Manager -- data synchronization

The image below represents an Everyplace Server domain where limited components deployed.


Everyplace Synchronization Manager
Customer Defined Services
Synchronization Traffic
Message Queueing Traffic (over PPP/Wireless Gateway Client)
Everyplace Wireless Gateway
MQSeries Everyplace
SecureWay Directory (LDAP)

Limited Everyplace Server installation with Wireless Gateway

WebSphere Everyplace Server, Service Provider Base Offering

Everyplace Server, Service Provider Base Offering opens the featured components, except the Everyplace Wireless Gateway and Everyplace Synchronization Manager. This deployment may be suitable for a company that does not require mobile data synchronization capabilities or connectivity solutions. The company may have a user support infrastructure in place. See "Content provider -- application service provider" for a description of this kind of enterprise.

Implementation Scenarios: Content provider -- application service provider
A new application services provider has been retained by a small business college to give its students and teachers access to programs from home and school. Teachers may enter the restricted student grading area while students are limited to using the programs placed on the server. An example of how communications flow within this Everyplace Server domain could be:

- WebSphere Application Server allows server sided applications to function on the host site.
- A user connects through Wireless Gateway, a third party gateway, or a traditional wired methods.
- The Edge Server Load Balancer then dispatches the request for authentication.
- WebSEAL-Lite performs the authentication which allows the user access.
- Policy Director verifies the user's abilities to enter restricted areas.
- Edge Server Caching Proxy uses the Active Session Table to verify subsequent interactions from the user across the domain.

Deployment Model
The scenario above requires the following Everyplace components.

- WebSphere Application Server -- enables web transactions and interactions

- WebSEAL-Lite -- user authentication and security
- Tivoli Personalized Management System -- subscriber and device management
- Edge Server Caching Proxy -- performance optimization
- Edge Server Load Balancer -- performance optimization

Everyplace Wireless Gateway for Everyplace Server, Service Provider Base Offering

Everyplace Wireless Gateway requires the installation of Everyplace Server, Service Provider Base Offering. This package includes featured components as well as Everyplace Wireless Gateway. This deployment could be typical for an enterprise that is looking for a solution for pervasive device support along with traditional wired internet services, such as an internet service provider or telephone company. An enterprise choosing this deployment prefers no off-line data synchronization capabilities. See "Internet service provider -- pervasive support" for this type of installation.

Implementation Scenario: Internet service provider -- pervasive support
A new internet service provider wants to specialize in pervasive device services. The provider requires users to subscribe in order to obtain the new service. An example of how communications flow within this Everyplace Suite domain could be:

- A registered user connects to the service with a pervasive device, and the Wireless Gateway forwards the request.
- The Edge Server Load Balancer then dispatches the request to WebSEAL-Lite.
- WebSEAL-Lite performs the authentication which allows the user access.
- Policy Director verifies the user's subscription to the service.
- The Edge Server Load Balancer dispatches the request to the appropriate WebSphere Transcoding Publisher server.
- The Transcoding Publisher forwards the request to the server handling the back end customer defined services (possibly the WebSphere Application Server).
- The Application Server returns the requested data to the Transcoding Publisher.
- The Transcoding Publisher reformats the requested data according to the cell phone's display specifications and returns it to the Wireless Gateway.
- The Wireless Gateway forwards the reformatted data to the user's pervasive device.

Deployment Model
This scenario requires the following Everyplace Suite components:

- Everyplace Wireless Gateway -- wireless device connectivity
- WebSEAL-Lite -- user authentication and security
- Tivoli Personalized Services Manager -- subscriber and device management
- WebSphere Transcoding Publisher -- data transcoding
- Edge Server Load Balancer -- performance optimization

Everyplace Synchronization Manager for Everyplace Server, Service Provider Base Offering

Everyplace Synchronization Manager requires the installation of Everyplace Server, Service

Provider Base Offering. The package opens all featured components in Everyplace Server, except Everyplace Wireless Gateway. This deployment could be typical for an enterprise that already has a wireless gateway or does not plan to support pervasive devices. The company has a device and user support infrastructure in place, including user and subscriber management, so a limited installation is acceptable. See "Enterprise customer -- appliance repair company" for a description of this kind of enterprise.

Implementation Scenario: Enterprise customer - appliance repair company

An appliance repair facility wants to update their appointment system. The company wants to equip its repair technicians with mobile devices to access daily schedules and record customer billing in the field. At the end of the day, the technicians reconnect to update the database with their notes. One possible example of how communications flow within this Everyplace Server domain could be:

- Tivoli Personalized Services Manager manages a Universal Database with customer information and appointments.
- Before beginning the day, a field employee equipped with a mobile device connects to the company server using a traditional wired method to download the day's schedule onto the device.
- The employee uses the device to refer to the schedule and make notes about each appointment.
- At the end of the day, the employee reconnects to the server.
- Everyplace Synchronization Manager synchronizes the Universal Database with the information entered by the employee.

Deployment Model

This scenario would require the following Everyplace components.

- Everyplace Synchronization Manager -- data synchronization
- Tivoli Personalized Services Manager -- subscriber and device management
- IBM DB2 Universal Database -- data storage

Related information

- Everyplace Server components
- External information

Requirements and prerequisites

WebSphere Everyplace Server has specific system requirements and component prerequisites that need to be understood prior to installation. Component prerequisites and automatic installation of components are discussed in this section.

- Hardware requirements
- Operating system support and requirements
- Disk space requirements
- Component software prerequisites and corequisites
- Installation data requirements
- Related information

Hardware requirements

The table below contains some hardware requirement information. You may want to refer to specific component documentation for more information.

Table 1. Hardware requirements

Component	Hardware requirements
WebSphere Everyplace Server base machine	● Operating system, see Operating system support and requirements below for more information ❍ AIX 4.3.3 Maintenance Level 8 ❍ Sun Solaris 7 and 8, SPARC-based systems ● Everyplace Server must be installed on an AIX or Sun Solaris X-Window workstation. ● See Disk space requirements and Component requirements and prerequisites below for more information about how much disk space is needed per machine. Also see the other components in this table for more information.

IBM Everyplace Wireless Gateway	● IBM eServer pSeries with 100 MB disk space available. See IBM Everyplace Wireless Gateway for additional information on how to determine storage and disk requirements. ● LAN adapter for connection to the IP network and to the radio network gateway (RNG) of either a DataTAC-TCP, Mobitex-TCP, or RNC3000 radio network. ● Network access to a simple mail transport protocol (SMTP) server. ● Additional hardware requirements specific to the network configuration, connectivity, and system based needs. See IBM Everyplace Wireless Gateway documentation for additional information on hardware requirements.
Tivoli Personalized Services Manager	● 2GB RAM, 2GB hard disk space (minimum system). See *Planning for Tivoli Personalized Services Manager* for detailed requirements. ● A separate volume ID is recommended. ● See Configure file systems for Tivoli Personalized Services Manager below for information about setting up files systems during initial installation of the operating system.
WebSphere Edge Server Caching Proxy	● 4GB Ultra SCSI disk or 16GB SSA disk
WebSphere Everyplace Suite Manager	● Runs on the supported levels of AIX and Solaris ● It also runs on Windows 2000

Operating system support and requirements

Everyplace Server runs on the following operating systems:

- AIX version 4.3.3 Maintenance Level 8 plus APARs
- Sun Solaris version 7 and 8

Some component administration consoles also run on Windows. See specific component documentation for further information.

AIX version 4.3.3 Maintenance Level 8 plus APARs

Before installing Everyplace Server on AIX, ensure that AIX version 4.3.3 Maintenance Level 8 is installed. If this level of modification has not been applied to your AIX system, ensure that the AIX Program Temporary Fixes (PTFs) or filesets noted below are applied before starting the installation.

- AIX Program Temporary Fixes
- PTFs for Oracle Database
- Java Runtime Environment
- APARs

AIX Program Temporary Fixes

The following PTFs are for all locales and are not on the AIX 4.3.3 installation media. They can be obtained from IBM if they are not already on your AIX system. To upgrade use the FixDist tool, available from: *http://techsupport.services.ibm.com/rs6000/support*.

***Note:** The PTFs shown with a "*" are at a higher level due to problems with performance, Netscape, and IBM HTTP Server failing to start when WebSphere Application Server is installed.

- bos.adt.include 4.3.3.10
- bos.adt.prof 4.3.3.53*
- bos.adt.samples 4.3.3.12
- bos.diag.com 4.3.3.13
- bos.diag.rte 4.3.3.13
- bos.diag.util 4.3.3.11
- bos.mp 4.3.3.16 if multiprocessor
- bos.net.ipsec.keymgt 4.3.3.10
- bos.net.nfs.client 4.3.3.10
- bos.net.tcp.client 4.3.3.15
- bos.net.tcp.server 4.3.3.14
- bos.rte 4.3.3.10
- bos.rte.aio 4.3.3.11
- bos.rte.control 4.3.3.10
- bos.rte.libc 4.3.3.53*
- bos.rte.libpthreads 4.3.3.51*
- bos.rte.methods 4.3.3.13
- bos.rte.net 4.3.3.1
- bos.rte.tty 4.3.3.10
- bos.sysmgt.serv_aid 4.3.3.13
- bos.sysmgt.trace 4.3.3.11
- bos.up 4.3.3.16 if uniprocessor
- devices.chrp.base.rte 4.3.3.12
- devices.common.base.diag4.3.3.10
- devices.isa_sio.baud.rte 4.3.2.1
- devices.ssa.disk.rte 4.3.3.10
- perfagent.tools 2.2.33.10
- X11.adt.lib 4.3.3.10
- X11.adt.motif 4.3.3.12
- X11.base.lib 4.3.3.15
- X11.base.rte 4.3.3.14
- X11.compat.lib.X11R5 4.3.3.10
- X11.Dt.lib 4.3.3.10
- X11.Dt.rte 4.3.3.10

Note: Do not use 4.3.3.27 - 4.3.3.50. These levels may cause performance problems in applications that create detached threads, for example IBM GSKit SSL Library.

- X11.motif.mwm 4.3.3.10
- X11.motif.lib 4.3.3.15

PTFs for Oracle Database

If you install Tivoli Personalized Services Manager with the Oracle Database, you must apply the following AIX filesets. They are located on the AIX V4.3.3 installation media:

- bos.adt (ALL)
- xlC.rte.* - C++ runtime library for IBM AIX (ALL)
- X11.adt (ALL)
- X11.base (ALL)
- perl.rte 5.5.3 (ALL)
- bos.compat.termcap - Termcap Compatibility Package (2)
- bos.sysmgt.trace 4.3.3.11
- devices.ssa.disk.rte 4.3.3.10
- Update all AIX components to AIX 4.3.3 Maintenance Level 02(IY06844) or higher

Java Runtime Environment 1.3

In addition, IBM AIX Developer Kit, Java 2 Technology Edition, Version 1.3.0, requires the following APARs be applied to your AIX system, if you are using these optional filesets and already have the base level filesets for specific locales or for double-byte character set (DBCS) locales. If they are not already installed, they can be found on the AIX 4.3.3 installation media or use the FixDist tool, available from: *http://techsupport.services.ibm.com/rs6000/support*.

- jkit.Wnn6.base 2.1.1.5
- bos.loc.com.JP 4.3.3.11
- bos.loc.utf.ZH_TW 4.3.3.11
- devices.rsa_sio.baud.rte 4.3.2.1
- X11.fnt.fontserver 4.3.3.12
- X11.fnt.ucs.ttf (for ja_JP or Ja_JP)
- X11.fnt.ucs.ttf_CN 4.3.3.1 (for zh_CN or Zh_CN)
- X11.fnt.ucs.ttf_KR (for ko_KR)

- X11.fnt.ucs.ttf_TW (for zh_TW or Zh_TW)

APARs

IBM HTTP Server does not start after installing WebSphere Application Server: APAR IY19277 or efix contained in 22869.379.000.efix.010613.01.tar

If you are using AIX 4.3.3.0 Maintenance Level 07 or higher, after WebSphere Application Server is installed, IBM HTTP Server does not start unless the machine is rebooted. If IBM HTTP Server is stopped, it does not restart until the machine is rebooted. To fix this problem, open a PMR with IBM against AIX, request APAR IY19277 or efix contained in 22869.379.000.efix.010613.01.tar. After September 5, 2001, this fix will be available from http://service.software.ibm.com/rs6k/fixdb.html.

Sun Solaris version 7 and 8

Install Sun Solaris 7 or 8 Entire Distribution Plus OEM Support and install all recommended patches. To get the most recent recommended patches go to: http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access. It is recommended that the install be done in single session mode to do the upgrade.

Tivoli Personalized Services Manager

Tivoli Personalized Services Manager requires the following Solaris components be installed:

- SUNWarc
- SUNWbtool
- SUNWhea
- SUNWlibm
- SUNWlibms
- SUNWsprot
- SUNWtoo

If using Tivoli Personalized Services Manager with Oracle, after installing the necessary components, consult your Oracle 8.1.7 Installation Guide on what changes are needed to update the system variables:

1. Add these additional lines to `/etc/system` for machines with 512 MB RAM:

```
set msgsys:msginfo_msgmax = 65535
set msgsys:msginfo_msgmnb = 65535
set msgsys:msginfo_msgmap = 258
set msgsys:msginfo_msgmni = 256
set msgsys:msginfo_msgssz = 50
set msgsys:msginfo_msgtql = 1024
set msgsys:msginfo_msgseg = 32767
set shmsys:shminfo_shmmax = 536870912
set shmsys:shminfo_shmseg = 50
set shmsys:shminfo_shmmni = 300
set semsys:seminfo_semmni = 1024
set semsys:seminfo_semmap = 1026
set semsys:seminfo_semmns = 2048
set semsys:seminfo_semmnu = 2048
```

Note: If your machine has greater than 512 MB RAM, calculate the value for `shmsys:shminfo_shmmax` by calculating 90 percent of the physical memory in bytes, comparing the number to the number for `shmsys:shminfo_shmmax`, and using the higher value.

2. Reboot the system.

Disk space requirements

The tables below shows the minimum disk space requirements for the Everyplace Server components and key directories. The amount of disk space needed depends on which components are installed. The following disk space requirements needed for component installation are for the `/usr` directory on AIX systems and the `/opt` directory for Solaris systems.

Table 2. Disk space requirements

Local Directory	Disk space requirements
/ (AIX only)	40MB
/ (Solaris only)	140MB
/usr (AIX only)	Total MB for all components being installed
/opt (AIX only)	100MB If installing WebSphere Edge Server Caching Proxy
/opt (Solaris only)	Total MB for all components being installed
/db	1500MB
/home (AIX only)	100MB
/tmp	100MB
/var	50MB
/var/adm/logs (Tivoli Personalized Sevices Manager only)	30MB

Component software requirements and prerequisites

The following table displays the system requirements and prerequisites for each component. For additional prerequisite information, see the specific component documentation.

Note: Required software marked with a "*" must be installed and configured, unless optional, on the machine or in the domain prior to installing the component it is associated with. All other software mentioned is installed by Setup Manager.

Table 3. Component software prerequisites and corequisites

Everyplace Server feature component	Software prerequisites and corequisites	Total required disk space options
Everyplace Active Session Table (11MB, 400bytes per concurrent user)	● None	● 11MB plus 400bytes per concurrent user for cache files. For example, if you plan on having 100,000 users logged on concurrently, allocate an additional 40MB for cache files.
Everyplace Cookie Proxy (10MB)	● WebSphere Edge Server Caching Proxy (100MB) A dedicated instance of Caching Proxy is required. Cookie Proxy and WebSEAL-Lite cannot coexist using the same Caching Proxy. In order to run both on the same machine two instances of Caching Proxy are necessary and each instance must listen on a separate port.	● 110MB

Everyplace Intelligent Notification Services (7MB)	● *SecureWay Directory must be installed and running in the domain before installing this component. If you want to install SecureWay Directory and this component on the same machine, install SecureWay Directory first, then restart Setup Manager to install this component. Therefore, when installing this component, select to retrieve existing information from SecureWay Directory. ● DB2 Universal Database (Server 350MB/Client 150MB) ● WebSphere Application Server (100MB) ● IBM HTTP Server (40MB) ● Everyplace Wireless Gateway (250MB) is strongly recommended. User customization is required to use other gateways.	● 297MB with remote DB2 database ● 497MB with local DB2 database
Everyplace Location Based Services (15MB)	● *SignalSoft must be set up in the domain ● *Tivoli SecureWay Policy Director (64MB)	● 179MB
Everyplace Suite Manager (20MB)	● *SecureWay Directory must be installed and running in the domain before installing this component. If you want to install SecureWay Directory and this component on the same machine, install SecureWay Directory first, then restart Setup Manager to install this component. Therefore, when installing this component, select to retrieve existing information from SecureWay Directory. ● Java Runtime Environment 1.3 (30MB) ● *Netscape Navigator or Communicator to view documentation	● 150MB

Everyplace Synchronization Manager (190MB)	● DB2 Universal Database (Server 350MB/Client 150MB) ● *Lotus Notes/Domino 5 server (on local machine) if synchronizing with Lotus Notes (UNIX server) (see product documentation for disk space needs) ● *Microsoft Exchange Server 5.5 (Windows NT 4 or Windows 2000 server) if synchronizing with Microsoft Exchange (see product documentation for disk space needs)	● 340MB with remote DB2 database plus disk space for optional software ● 540MB with local DB2 database plus disk space for optional software
Everyplace Wireless Gateway (Gateway 250MB, Gatekeeper 50MB)	● *Tivoli SecureWay Policy Director (optional) (64MB) ● DB2 Universal Database (Server 350MB/Client 150MB) or Oracle8i database version 8.1.7 ❍ If using Oracle8i: Meramt DataDirect Connect ODBC 3.6.0 ❍ The client must be local, but the server can be remotely installed for either DB2 or Oracle. ● IBM GSKit SSL Library	● 400MB using Gateway with remote DB2 database ● 600MB using Gateway with local DB2 database ● 200MB using Gatekeeper with remote DB2 database ● 400MB using Gatekeeper with local DB2 database
MQSeries Everyplace (40MB)	● Java Runtime Environment 1.3 (30MB)	● 70MB
Sametime Everyplace	● Windows NT or Windows 2000 ● Sametime Connect 1.5 or higher ● Sametime 2.0 ● Domino Server ● Mobile Services for Domino ● A WAP gateway	
SecureWay Directory (100MB)	● DB2 Universal Database Server and Client on local machine (Server 350MB/Client 150MB) ● IBM HTTP Server (40MB) ● You also need 150MB additional free space in the /home directory	● 790MB

Tivoli Personalized Services Manager (Subscription Manager 75MB, Device Manager 45MB)	● DB2 Universal Database (Server 350MB/Client 150MB) or Oracle database (see product documentation for disk space needs) ❍ See Using Oracle for more information ● WebSphere Application Server (100MB) ● IBM HTTP Server (40MB) ● *JDK 1.2.2 ❍ On Solaris: must be installed before installing this component ❍ On AIX: installed with component	● 290MB if using remote DB2 database ● 490MB if using a local database
Tivoli SecureWay Policy Director (64MB)	● *IBM DCE (Distributed Computing Environment) version 3.1 for AIX and Solaris (150MB) ● SecureWay Directory (100MB) installed in the domain	● 314MB
Voice Services	● *IBM WebSphere Voice Server 1.5 for Windows ● *IBM Direct Talk for Windows or AIX ● *IBM ViaVoice for Windows Standard Edition	
WebSEAL-Lite (30 MB)	● *Tivoli SecureWay Policy Director (optional) (64MB) If using WebSEAL-Lite with Policy Director, follow the pre installation steps in Prepare for WebSphere Everyplace Server installation. ● WebSphere Edge Server Caching Proxy (100MB) A dedicated instance of Caching Proxy is required. WebSEAL-Lite and Cookie Proxy cannot coexist using the same Caching Proxy. In order to run both on the same machine two instances of Caching Proxy are necessary and each	● 136MB ● 200MB if using Policy Director

	instance must listen on a separate port. ● Everyplace Active Session Table (6MB)	
WebSphere Edge Server Caching Proxy (100MB)	● WebSphere Edge Server Load Balancer -- administration package and device driver (60MB) ● IBM GSKit SSL Library	● 160MB
WebSphere Edge Server Load Balancer (60MB)	● WebSphere Edge Server Caching Proxy -- if installing Content Based Routing (CBR) subcomponent (100MB)	● 60MB ● 160MB with Content Based Routing
WebSphere Transcoding Publisher (80MB)	● *SecureWay Directory must be installed and running in the domain before installing this component. If you want to install SecureWay Directory and this component on the same machine, install SecureWay Directory first, then restart Setup Manager to install this component. Therefore, when installing this component, select to retrieve existing information from SecureWay Directory. ● *JDK 1.2.2 ○ On Solaris: must be installed before installing this component ○ On AIX: installed with component	● 80MB

Table 4. Component database management support

Component	DB2 Universal Database	Oracle
SecureWay Directory	Yes	Not supported
Everyplace Intelligent Notification Services	Yes	Not supported
Everyplace Synchronization Manager	Yes	Yes. Oracle must be installed and configured before installing this component.

Everyplace Wireless Gateway	Yes	Yes. Oracle must be installed and configured before installing this component.
Tivoli Personalized Services Manager	Yes	Yes. Oracle must be installed and configured before installing this component.
WebSphere Application Server	Yes	Yes. Oracle must be installed and configured before installing this component.

Installation data requirements

Setup Manager prompts you for different types of information depending on the components you choose to install. See the list of components and types of information in the table below. Be prepared to provide the following types of information at installation time:

Table 5. Information asked during installation

Component	User information	LDAP information	Database information	Local server information
DB2 Universal Database	x	x	x	x
Everyplace Active Session Table	x (group not asked)	x	x	x
Everyplace Intelligent Notification Services	x		x	x
Everyplace Location Based Services	x (group not asked)	x	x	x
IBM HTTP Server	x			x
SecureWay Directory	x	x	x	x
Tivoli Personalized Services Manager	x			x
WebSEAL-Lite	x			
WebSphere Application Server	x		x	x
WebSphere Edge Server Caching Proxy	x			x

WebSphere Edge Server Load Balancer	x			x
WebSphere Transcoding Publisher	x (not group)			x

Table 6. Definition of each information type above

Information types			
User	**LDAP**	**Database**	**Local server**
● User ID ● Group ● Password **Note:** Use only lower case letters (a-z) or numbers (0-9) for the user ID, group, and password.	● Server name ● Directory suffix ● Object type ● Database name ● Database instance ● Database home directory ● LDAP Administrator user ID ● LDAP Administrator Password ● Port	● Database name ● Database instance ● Database home directory ● Port	● Host name ● Domain name ● TCP/IP address

Related information

- Everyplace Server components
- External information

Install Oracle Database for Tivoli Personalized Services Manager

If you plan to use Oracle Database software for your database management system instead of IBM DB2 Universal Database, follow the instructions in this section. Oracle must be installed prior to installing Tivoli Personalized Services Manager.

The following section comes from *Installing Tivoli Personalized Services Manager* and outlines the steps necessary to install and configure Oracle for proper installation of Tivoli Personalized Services Manager. This installation procedure applies to installation for both IBM AIX and Sun Solaris. For complete installation instructions, refer to the Oracle 8.1.7 Installation Guide.

1. If using AIX go to the next step. If using Solaris, before installing Oracle modify your `/etc/system` file after you have consulted the Oracle installation documentation. In most cases you need to add additional lines to the system file depending on how much physical RAM you have on your machine.
2. Create the Oracle product home directory using the following command:
   `mkdir -p /db/app/oracle/products/8.1.7/OraHome1`
3. Create the **dba** group.
4. Create the oracle8 user with the password oracle.
5. Make **oracle8** a member of the **dba** group.
6. Make the **oracle8** home the directory:
   `/db/app/oracle`
7. Create a `.profile` file in the **oracle8** home directory:
   `/db/app/oracle`
8. Verify that the `.profile` file contents are executable using the following command:
   `chmod 755 .profile`
9. Verify that the `.profile` file contains the following lines to ensure that the schema installation will not fail:

   ```
   #!/bin/ksh
   PATH=/usr/ccs/bin:$PATH:/usr/local/bin
   LIBPATH=:/usr/ccs/lib
   LD_LIBRARY_PATH=:/usr/ccs/lib
   DISPLAY=localhost:0.0
   export DISPLAY
   #
   ORACLE_SID=ispb
   export ORACLE_SID
   ```

```
ORACLE_BASE=/db/app/oracle
export ORACLE_BASE

ORACLE_HOME=/db/app/oracle/products/8.1.7/OraHome1
export ORACLE_HOME

ORACLE_OWNER=oracle8
export ORACLE_OWNER

PATH=$ORACLE_HOME/bin:$PATH
LIBPATH=$ORACLE_HOME/lib:$LIBPATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH

export LIBPATH
export LD_LIBRARY_PATH
export PATH
```

Note: LIBPATH is needed on IBM AIX and LD_LIBRARY_PATH is needed on Sun Solaris, but including both on either system will not damage the installation process.

10. Run the command **xhost** + localhost on the system where you will run the database installation program.
11. If telneting to the Oracle server or Oracle client machine from your machine:
    . Run the command **xhost** + localhost on your machine terminal.
    b. Run the following command from the telnet session: **export DISPLAY=MachineX:0.0** where MachineX is the name of your machine.
12. Refer to the Oracle 8.1.7 Installation Guide and complete all recommended installation tasks. On the Oracle GUI installer screens, you will be asked to enter product information.
    - For the Database server, use the following information as a guide:
        - For the Oracle home path, specify the following: `/db/app/oracle/products/8.1.7/OraHome1`
        - When prompted, select **Enterprise Server**.
        - When prompted, select **Typical Install**.
        - Enter a name for the Global Database Name and Oracle SID.
        - For the location of the database files, specify the following: `/db/dbfiles`
        - When prompted, select **Yes** for creating a starter database.
        - Accept the default for user local bin directory, that is: `/usr/bin/local`
        - Use Net8 Configuration Assistant to specify the following information:

    - – Do *not* select **Typical**.
      – When prompted, select **No** for configuring Directory services.
      – For protocols, accept the defaults.
      – Do not configure another listener.
      – Use the standard port number.
      – When prompted, accept the default No, **do not change naming methods**.
    - Log on to and verify the sample database.
    - Run the command **dbassist** and follow the instructions to delete the database.
- For the Database Client, use the following information as a guide:
    - For Oracle home path, specify: `/db/app/oracle/products/8.1.7/OraHome1`
    - For installation type, select **Oracle Client**.
    - For the installation option, select **Application User**.
    - When prompted, select **Delay Service Directory Install**.
    - When prompted, select **Oracle8i**.
    - When prompted, assign **Service Name** as the previously created database name.
    - Select a database server host.
    - To test the Oracle install, login to the default database, if you created it at install time, using sqlplus as user oracle8. If you are able to use sqlplus odds are that the Oracle install was successful. Remove the default database instance if its name is identical to the name that you have chosen to use for your Tivoli Personalized Services Manager database instance after you are done testing.

Note:

- It is highly recommended that you complete all post-installation steps as outlined in the *Oracle 8.1.7 Installation Guide*.
- When you configure Net8 client on a machine that is not on the server, its Net Service name should be something different then what the database name is that you are trying to connect to. For example, `dbname=ispb`, `NetService name=tsmdb`.

XML automated installation

What is XML automated installation?

XML automated installation helps to automate the installation of Everyplace Server. During initial install, the Setup Manager panels can be initialized with information retrieved from a *planner.xml* file. When a *planner.xml* file is supplied, Setup Manager will validate the file and use it to fill in the required input fields.

The key components of XML automated installation are an editable *planner.xml* file and an associated *planner.dtd* file. Advanced Knowledge of XML is not required to use the XML automated installation but you should have some proficency with XML and be comfortable editing and making changes to an XML file.

There are several basic steps to using the XML automated installation.

1. Edit *planner.xml*.
2. Run Everyplace Server Setup Manager.
3. In Setup Manager, select the subset of configuration information to use.

XML automated installation provides the following benefits.

1. The *planner.xml* file can contain all the editable values entered during the process of an installation, such as LDAP connection, components to install, and user IDs.
2. The *planner.xml* file is not OS dependent. Multiple host configurations can be created for installation across different computers and platforms.
3. Installation information can be collected prior to installation and placed in the *planner.xml* file, reducing conflicts during installation.
4. The *planner.xml* file can help eliminate repetative entry steps from identical installations.

Editing the *planner.xml* file

1. Either copy `/tmp/IBMEPS/planner.xml` to a local subdirectory of your choice, or edit the file in place. The file location does not matter, as long as the file browser in Setup Manager can locate the *planner.xml* file.

2. Edit *planner.xml* and configure the parameters you want to use during the Everyplace Server installation. Define multiple hosts if desired. Use a text editor to edit the planning file. Refer to the *planner.dtd* file listed below for a complete definition of the required fields and structuring of the *planner.xml* file.

   Note: No passwords should be entered in the *planner.xml* file.

Note: Unless otherwise stated, everything in quotes is case sensitive.

3. Validate the modified *planner.xml* with the *planner.dtd* file. If the *planner.xml* file does not properly conform to the *planner.dtd*, some fields may not be properly filled in during installation, and unpredictable outcomes may result.

The *planner.xml* file lets you define common configuration settings and host-specific configuration settings to override the common settings. Setup Manager reads in the common configuration settings first and uses them as default settings for all XML automated installations.

Common configuration settings are defined within the `<common></common>` tags.

Host-specific configuration settings are defined within the `<host></host>` tags, using the `name=` attribute to define the hostname. For example, the opening tag, `<host name=machine.this.place.com>` would delineate configuration settings to be used for automated installation on the host, `machine`, in the domain, `this.place.com`.

WARNING: DO NOT MODIFY the *planner.dtd* file. Use the *planner.dtd* file as a reference for properly modifying the *planner.xml* file. Also, be sure to use *planner.dtd* to validate the modified *planner.xml* before attempting to use it with Setup Manager. If the *planner.xml* file does not properly conform to the *planner.dtd*, some fields may not be properly filled in during installation, and unpredictable outcomes may result.

Using the *planner.xml* file with Everyplace Server Setup Manager

1. Run Everyplace Server Setup Manager. For more information on how to run Everyplace Server Setup Manager, refer to the InfoCenter section about Everyplace Server Setup Manager.
2. Everyplace Server Setup Manager prompts for the location of XML automated installation file. Use the file browser to locate the *planner.xml* file.
3. Setup Manager performs a simple XML validation. If there are any syntax errors in the *planner.xml* file, you will prompted to fix the file and retry Everyplace Server installation.

   Note: Be sure to validate *planner.xml* against *planner.dtd*. If the *planner.xml* file does not properly conform to the *planner.dtd*, some fields may not be properly filled in during installation, and unpredictable outcomes may result.
4. Setup Manager loads the common configuration settings. Then Setup Manager attempts to locate the host-specific configuration settings that match the hostname of the current computer. Setup Manager searches *planner.xml* for a `<host>` tag with the `name` attribute value that matches the host name of the current computer. If Setup Manager does not find such a `<host>` tag, it will display a selection panel with a list of alternative host names that are defined in the *planner.xml* file.
5. Select one of the alternative host names listed, if you wish to use the configuration settings defined for that host. If you wish to use only the common configuration settings as defined in the *planner.xml* file, select None.

Sample *planner.xml* file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE INSTALL SYSTEM "planner.dtd">
```

```
<INSTALL>
    <TITLE VALUE="DISTRIBUTED SETUP 1A" />
    <LICENSE VALUE="XXXX-XXXX-XXXX-XXXX-XXXX" />
    <LDAP FILE="" OPTION="RETRIEVE">
        <LDAP_LOGIN PORT="389" PW="" SERVER="ldap.host.ibm.com" UID="ldap" />
    </LDAP>
    <COMMON>
        <AUTH>
            <MULT_AUTH NAME="mqe" UID="user" GID="group" />
        </AUTH>
        <PACKAGE MODE="INSTALL" NAME="mqe" VERSION="1" />
    </COMMON>
    <HOST NAME="host1.domain.ibm.com">
        <AUTH>
            <MULT_AUTH NAME="wep" UID="wepg" GID="wepg" />
            <MULT_AUTH NAME="ins" UID="ins" GID="insg" />
        </AUTH>

        <DBMS PREFERRED="oracle">

                <DB NAME="db2" REMOTE="YES">
              <PARAM NAME="hostname" VALUE="db2.host.ibm.com">
                <ERROR ACTION="TRY" VALUE="db2a.host.ibm.com" />
              </PARAM>
            </DB>
                <DB NAME="oracle" REMOTE="YES">
              <PARAM NAME="hostname" VALUE="ora.host.ibm.com" />
                </DB>
        </DBMS>
        <PACKAGE NAME="wep">
            <PARAM NAME="proxymode" VALUE="TP mode" />
            <PARAM NAME="primaryast" VALUE="prim_ast.ibm.com" />
            <PARAM NAME="secondaryast" VALUE="sec_ast.raleigh.ibm.com" />
        </PACKAGE>
        <PACKAGE NAME="ins">
            <PARAM NAME="iqueueservice" VALUE="false" />
            <PARAM NAME="undservice" VALUE="false" />
            <PARAM NAME="scsservice" VALUE="true" />
        </PACKAGE>
    </HOST>
    <HOST NAME="host2.host.ibm.com">
        <AUTH>
            <COMMON_AUTH UID="user" GID="group" />
        </AUTH>
        <PACKAGE NAME="lbs">
            <PARAM NAME="pdurifile" VALUE="/home/user/url.txt" />
            <PARAM NAME="pdldapserver" VALUE="ldap.host.ibm.com" />
            <PARAM NAME="pdldapuser" VALUE="cn=root" />
            <PARAM NAME="pdldapport" VALUE="1026" />
            <PARAM NAME="locationserverport" VALUE="2000" />
            <PARAM NAME="locationserver" VALUE="sigsoft.host.ibm.com" />
        </PACKAGE>
        <PACKAGE NAME="ewg">
            <COMPONENT NAME="ewg_svr" />
            <COMPONENT NAME="ewg_kpr" />
            <COMPONENT NAME="ewg_ard" />
            <COMPONENT NAME="ewg_tac" />
            <COMPONENT NAME="ewg_rad" />
            <COMPONENT NAME="ewg_dal" />
            <COMPONENT NAME="ewg_lan" />
            <COMPONENT NAME="ewg_mob" />
            <COMPONENT NAME="ewg_src" />
            <COMPONENT NAME="ewg_rnc" />
        </PACKAGE>
     </HOST>
</INSTALL>
```

planner.dtd file

```
<?xml encoding="UTF-8"?>
<!ENTITY % ldapvalues           "(INSTALL|RETRIEVE|FILE)">
<!ENTITY % yesno                "(YES|NO)">
<!ENTITY % dbtype               "(oracle|db2)">
<!ENTITY % denied_action      "(TRY|DEFAULT|ABORT)">
<!ENTITY % valid_install_mode "(INSTALL|UNINSTALL)">
<!ENTITY % valid_package_name
"(wte|wep|lbs|ins|nd|ast|wec|esm|ewg|mqe|swd|tsm|wtp)">
<!ELEMENT INSTALL (TITLE?,LICENSE,LDAP,COMMON?,HOST*)>

<!ELEMENT TITLE EMPTY>
<!ATTLIST TITLE VALUE CDATA #IMPLIED>

<!ELEMENT LICENSE EMPTY>
<!ATTLIST LICENSE VALUE CDATA #REQUIRED>

<!ELEMENT HOST (AUTH,DBMS?,PACKAGE*)>
<!ATTLIST HOST NAME CDATA #REQUIRED>

<!ELEMENT PACKAGE (COMPONENT*,PARAM*)>
<!ATTLIST PACKAGE NAME %valid_package_name; #REQUIRED>
<!ATTLIST PACKAGE VERSION CDATA #REQUIRED>
<!ATTLIST PACKAGE MODE %valid_install_mode; #REQUIRED>

<!ELEMENT COMPONENT EMPTY>
<!ATTLIST COMPONENT NAME CDATA #REQUIRED>

<!ELEMENT COMMON (AUTH,PACKAGE*)>
<!-- kind of like a host -->

<!ELEMENT AUTH (COMMON_AUTH|MULT_AUTH+)>
<!-- uid/gid, common or different for each package -->

<!ELEMENT COMMON_AUTH EMPTY>
<!ATTLIST COMMON_AUTH UID CDATA #REQUIRED>
<!ATTLIST COMMON_AUTH GID CDATA #REQUIRED>
<!ATTLIST COMMON_AUTH PWD CDATA #IMPLIED>
<!-- If not present then prompt? -->
<!-- or maybe just use a default -->
```

```
<!ELEMENT MULT_AUTH EMPTY>
<!ATTLIST MULT_AUTH NAME CDATA #REQUIRED>
<!ATTLIST MULT_AUTH UID CDATA #REQUIRED>
<!ATTLIST MULT_AUTH GID CDATA #REQUIRED>
<!ATTLIST MULT_AUTH PWD CDATA #IMPLIED>

<!ELEMENT PARAM (ERROR?)>
<!ATTLIST PARAM NAME CDATA #REQUIRED>
<!ATTLIST PARAM VALUE CDATA #IMPLIED>

<!ELEMENT ERROR (ERROR?)>
<!ATTLIST ERROR ACTION %denied_action; #REQUIRED>
<!ATTLIST ERROR VALUE CDATA #IMPLIED>

<!ELEMENT LDAP (LDAP_LOGIN?)>
<!ATTLIST LDAP OPTION %ldapvalues; #REQUIRED>
<!ATTLIST LDAP FILE CDATA #IMPLIED>
<!-- if option 1 is hit, then swd must be an install component to get params-->

<!ELEMENT LDAP_LOGIN EMPTY>
<!ATTLIST LDAP_LOGIN SERVER CDATA #IMPLIED>
<!ATTLIST LDAP_LOGIN UID CDATA #IMPLIED>
<!ATTLIST LDAP_LOGIN PORT CDATA #IMPLIED>
<!ATTLIST LDAP_LOGIN PWD CDATA #IMPLIED>
<!-- all implied so that only specific entries have to override defaults -->

<!ELEMENT DBMS (DB+)>
<!ATTLIST DBMS PREFERRED %dbtype; #REQUIRED>

<!ELEMENT DB (PARAM?)>
<!ATTLIST DB NAME %dbtype; #REQUIRED>
<!ATTLIST DB REMOTE %yesno; #REQUIRED>
```

XML tags for *planner.xml*

Tag Name	Values	Description
DOCTYPE	"/tmp/IBMEPS/planner.dtd"	Logical path to planner.dtd file.
TITLE VALUE=	VALUE="Distributed setup 1A"	XML automated installation file title and version level.
LICENSE VALUE=	VALUE="xxxx-xxxx-xxxx-xxxx-xxxx"	License key information.

LDAP
FILE=
OPTION=
	FILE="/tmp/ladap.ldif"
	OPTION=RETRIEVE
	OPTION=INSTALL
	OPTION=FILE
		FILE is the name of the LDIF file to write to.
		For the OPTION attribute, choose one of the three available options, listed to the right.

LDAP_LOGIN
PORT=
SERVER=
UID=
	PORT="389"
	SERVER="ldap.host.company.com"
	UID="ldapid"
		LDAP login information:
		PORT=Port number to connect to retrieve information from LDAP.
		SERVER=The hostname of the LDAP server.
		UID=the user ID of the ldap server.

COMMON	This tag contains configuration settings information that will be entered automatically into the relevant Setup Manager fields.	The <common> tags enclose default installation settings and values. Each setting and value defined here will be automatically entered into fields in the installation GUI, unless they are overidden by settings or values defined for a selected <host>.
HOST NAME=	NAME="host1.domain.company.com" The value for NAME should be a fully qualified Hostname so it will be unique.	The <host> tags enclose installation settings and values designated to override settings and values defined within the <common> tags. The <host> settings will override the <common> settings for installation on a host whose hostname matches the NAME value or when the hostname is chosen from a selection list in the installation GUI.

AUTH		<auth> tags enclose authentication information for components.

MULT_AUTH
NAME=
GID=
UID=
	NAME="wep" (component name)
	UID="user" (user ID)
	GID="group" (group ID)
		One or more MULT_AUTH tag can be specified within an AUTH tag set. A MULT_AUTH tag can only be defined if a COMMON_AUTH tag has not been defined. The NAME attribute denotes the component abbreviation for the desired package. Click here for a list of component names and abbreviations.

COMMON_AUTH
GID=
UID=
	UID="user" (user ID)
	GID="group" (group ID)
		One COMMON_AUTH tag can be defined within an AUTH tag set. A COMMON_AUTH tag can only be defined if a MULT_AUTH has not been defined. The NAME attribute is not used for this tag. No particular component needs to be specified, because the COMMON_AUTH tag defines authentication parameters for all components within an AUTH tag set.

DBMS PREFERRED=	PREFERRED="db2" This value defines which of the following defined databases is preferred.	The <dbms> tags enclose one or more database definitions. The PREFERRED attribute takes the NAME of the preferred database.

DB
NAME=
REMOTE=
	NAME="db2"
	REMOTE="YES"
	The value for the NAME attribute can be either "db2" or "oracle."
		A database definition, including attributes defining the name and whether or not the database is remote.

PARAM NAME="hostname"	VALUE="db2.host.domain.com"	Define a <param> within the <db> tags for the hostname of the database.
PACKAGE NAME=	NAME="wep" (component name)	Define a <package> to install that package. The NAME attribute denotes the component abbreviation for the desired package. Click here for a list of component names and abbreviations.
PARAM NAME= VALUE=	NAME="proxymode" (the name of an installation parameter for the specified component) VALUE="TP mode" (the value to enter automatically for the above named parameter)	Each package definition contains a set of parameters that are entered automatically for the package at installation time. Each parameter has a name and a value. Click here for a list of parameters and values.
COMPONENT NAME=	NAME="ewg_svr"	If a component defined in a PACKAGE tag set has installable subcomponents, one or more COMPONENT tags may be used to specify which subcomponents to install. Click here for a list of component and subcomponent names and abbreviations.

Everyplace Server component names and abbreviations

The following table lists Everyplace Server components and the corresponding abbreviations. Use the following component abbreviations in the *planner.xml* file, in the <package> tag, as values for the NAME attribute.

Component name	Component abbreviation	Subcomponents
WebSphere Edge Server, Caching Proxy	wte	ewg_svr (Gateway)
WebSEAL-Lite	wep	
Location Based Services	lbs	
Everyplace Cookie Proxy	ecp	
Universal Notification Dispatcher	nd	nd_bas (Dispatcher) nd_cbr (Content Based Routing) nd_iss (Interactive Session Support)
Everyplace Active Session Tables	ast	
Everyplace Intelligent Notification Services	ins	
Everyplace Suite Manager	wec	
Everyplace Synchronization Manager	esm	
Everyplace Wireless Gateway	ewg	ewg_svr (Gateway) ewg_kpr (Gatekeeper) ewg_ard (Ardis Support) ewg_tac (DataTAC Support) ewg_rad (Dataradio Support) ewg_dal (Dial

		Support) ewg_lan (IP LAN Support) ewg_mob (Mobitex Support) ewg_sms (Short Messaging Service Support) ewg_smt (SMTP Support) ewg_snp (SNPP support) ewg_src (Modacom-SRC Support) ewg_rnc (Motorola PMR Support)
MQSeries Everyplace	mqe	
SecureWay Directory Services	swd	
Tivoli Personalized Services Manager	tsm	tsmdbs (Database Integration) tsm_dms (Tivoli Device Manager) tsm_ens (Enrollment Server) tsm_ccs (Customer Care Support) tsm_mcs (Member Self Care Support) tsm_sms (System Management) tsm_wes (Everyplace Server Enabler) tsm_ptk (Portal Toolkit)
WebSphere Trascoding Publisher	wtp	

Everyplace Server component parameters

Parameters Currently Supported (Sorted by Component Name)

Component	Variables	Comments
SWD (SecureWay Directory Services)	dirsuffix objecttype dbname dbinstance ldapport	Suffix information to configure LDAP. Type of entries in LDAP. Name of the database to store entries for LDAP. Instance to use to add information to the database. The port this server will listen on.
TSM (Tivoli Personalized Services Manager)	dbport dbinstance dbuser dbhome numsubscribers webmaster	The port the remote db is listening on. The name of the db instance to use. The name of db User. If doing database integration, the location to store the database. The number of subscribers for TPSM. The email address of the TSM webmaster.
INS (Everyplace Intelligent Notification Services)	dbport dbname dbinstance dbhome iqueueservice undservice scsservice gbservice	The port the remote db is listening on. The name of db to use for INS. The db instance to use for INS. The home director of the db2 instance. IQueue Server. Universal Notification Dispatcher. Secure Context Server. Gryphon Broker.
LBS (Everyplace Location Based Services)	locationserver locationserverport pdldapserver pdldapuser pdldapport pdurifile	The server that LBS will be configured to talk to. The port that SignalSoft is listening on for LBS. The Policy Director LDAP Server. The user to login to the LDAP Server as, has a default:sec_master. The port LDAP is listening on. The location of a file that contains a list of Application(URIs) to load.
WEP (WebSEAL-Lite)	primaryast secondaryast proxymode defaultrealm	Hostname of Primary AST server. Hostname of the Secondary AST Server. Webseal Proxy Mode: AP or TP. The default Realm of WebSEAL.
WAS (WebSphere Application Server)	dbname dbport	The name of the database to user for Application Server. The port on which the database for Application Server is running..

Related information

- Everyplace server Setup Manager

Prepare for WebSphere Everyplace Server installation

Installing WebSphere Everyplace Server is made easy with the Everyplace Setup Manager. Setup Manager prompts you for all the necessary information to complete the installation. You can perform a local or a remote installation with Setup Manager. Setup Manager can also be run in a debug mode.

Before installing DB2 on Solaris

Solaris does not store or create user and group ids in numerical order. During installation, DB2 creates new user and group ids based on the last number available. In order to ensure that the ids are associated with unique numbers, make sure there are no gaps in the user and group id numbers and that the last id is the highest number used. You can create place holder ids to fill in the gaps and create a last entry that uses a number higher than the rest.

Prepare to install WebSEAL-Lite with Policy Director

To avoid errors when installing WebSEAL-Lite with Policy Director, perform the following steps to replace a library file before starting Setup Manager.

1. On the machine you want to install WebSEAL-Lite, quit all Policy Director applications.
2. On AIX, execute `slibclean` command to clean up the shared memory.
3. Make a backup copy of the `/opt/PolicyDirector/lib/libpdauthzn.a` file.

4. Mount Everyplace Server Disc 10, and locate the appropriate file depending on your operating system.
   `/PolicyDirector/aix/libpdauthzn.a`
   `/PolicyDirector/sun/libpdauthzn.so`
5. Copy the same file from Disc 10 to `/opt/PolicyDirector/lib`. Perform the same procedure on Solaris, except that the library have a suffix `.so`.

WebSphere Everyplace Server installation program: Setup Manager

Everyplace Server contains many individual components and requires several additional server products. Installing so many components individually is not practical, let alone installing them into a distributed setting. Intuitively, the Everyplace Server installation is designed to be a centralized installation of all components.

Many of the components within Everyplace Server are already mature products with their own installation process. The centralized Everyplace Server installation still utilizes the server software installation and upgrade mechanism already in place for the server components. The server installation is a coordinated effort. It determines what common configuration parameters can be "factored out" and supplied only once, but used by multiple components.

In addition, as components are installed and configured on specific systems, the Everyplace Server installation process captures configuration parameters and stores the information in LDAP for subsequent usage later in the installation process. This way, the parameters entered during installation can be saved for other Everyplace Server component installations.

Installation of Everyplace Server is made easy with Setup Manager. Setup Manager can perform the following tasks:

- Set up the Everyplace Server environment
- Allow limited migration from previous Everyplace Suite releases version 1.1.2 and version 1.1.3 to version 2.1.
- Help you save time on subsequent installations of Everyplace Server by allowing you to pre-define component information that is used to automatically populate fields during the installation.
- Provide a mechanism to validate that the installation and configuration completed successfully.

Begin local installation from product CDs

Be sure to exit all running programs prior to starting Everyplace Server installation. To start the Everyplace Server installation from the product CD, follow these steps:

1. Log in as root.
2. Place Disc 1 in the CD drive.
3. Mount the CD.
    - In the root directory, create the directory `/cdrom` (if it does not exist) by entering the command:

      ```
      mkdir /cdrom
      ```

      Note: Enter: `cd /` to get to the root directory. Be sure to issue these commands from the root directory. If any command window is in the `/cdrom` directory, you are not able to unmount the CD.
    - For AIX, enter the command:

      ```
      mount -rv cdrfs /dev/cd0 /cdrom
      ```

      To remove the CD, enter the command:

      ```
      umount /cdrom
      ```

      or click the **Unmount** button when prompted by Setup Manager.
    - On Solaris, the CD automatically mounts and appear in the file manager.
4. Enter the following command:
    - For AIX:

      ```
      /cdrom/install.sh
      ```

    - For Solaris:

      ```
       /cdrom/cdrom0/install.sh
      ```

5. Setup Manager requires Java 1.3. If Setup Manager does not locate Java 1.3 on the machine, you are prompted to install or locate Java 1.3, or exit Setup Manager. Enter the appropriate option to proceed.

    . Install Java 1.3 provided on your CD.

    b. Specify the location of your Java 1.3 JAVA_HOME path.

    c. Search the file system for Java 1.3 (This may take a long time).

    d. Quit the installation.
6. You are asked if you are installing Tivoli Personalized Services Manager. If so, the IBM AIX Developer Kit, Java 2 Technology Edition, version 1.2.2 is installed. If installing Tivoli Personalized Services Manager on Solaris, Java 1.2.2 must be installed before starting the installation.

Solaris Tip	If you click in the background area outside the install window, the install window goes to the background and may not be visible. To bring the install window back in view, press the Alt-Tab keys simultaneously until you see the window again.

Begin remote installation from file system

Use these instructions only if you are installing directly from a local or network file system instead of the product CDs. Be sure to exit all running programs prior to starting Everyplace Server installation. To start the Everyplace Server installation program from the file system, follow these steps:

1. Go to the directory for Disc 1 by entering:

   ```
   cd cd1name
   ```

   Where `cd1name` is the name of the directory containing the files from Disk 1.
2. Enter the following command to start the installation program:

   ```
   ./install.sh
   ```
3. Setup Manager uses Java 1.3. If Setup Manager does not locate Java 1.3 on the machine, you are prompted to install or locate Java 1.3, or exit Setup Manager. Enter the appropriate option to proceed.

   . Install Java 1.3 provided on your CD.

   b. Specify the location of your Java 1.3 JAVA_HOME path.

   c. Search the file system for Java 1.3 (This may take a long time).

   d. Quit the installation.
4. You are asked if you are installing Tivoli Personalized Services Manager. If so, the IBM AIX Developer Kit, Java 2 Technology Edition, Version 1.2.2 is installed. If installing Tivoli Personalized Services Manager on Solaris, Java 1.2.2 must be installed before starting the installation.

Run Setup Manager in debug mode

If you want to run Setup Manager in debug mode, perform the following steps before invoking the `install.sh` script. The table below contains all the available debug levels.

- For AIX:

  ```
  export IBMEPS_DEBUG_LEVEL=<level>
  ```
- For Solaris:

  ```
  IBMEPS_DEBUG_LEVEL=<level>
  ```

  ```
  export IBMEPS_DEBUG_LEVEL
  ```

Level Description	Level
DEBUG_LEVEL_OFF	0
DEBUG_LEVEL_MIN	1
DEBUG_LEVEL_INFO	2
DEBUG_LEVEL_MED	3
DEBUG_LEVEL_VERBOSE	4
DEBUG_LEVEL_ALL	5

Setup Tips

- Plan for the installation. A diagram or flow chart of your network may help.
- Install the components one by one and in layers.
- SecureWay Directory should be installed before any other component.
- Check for errors after each component is installed.
- Install components on one machine before installation on another machine begins.

Related information

- Installation steps
- CD contents

Installation steps

This section details the steps necessary to install WebSphere Everyplace Server in an environment with multiple machines.

Before installing Everyplace Server, be sure to read the Readme file for late-breaking installation news and tips. The Readme file can be viewed by starting the installation program and choosing the option to view the Readme file. It can also be found in the `/info` directory on Disc number 1 in HTML format.

- Overview
- Step one: Install SecureWay Directory
- Step two: Install Tivoli Personalized Services Manager
- Step three: Install other Everyplace Server components
- Related information

Overview

It is important to follow these instructions when installing the Everyplace Server components. Do not follow the component specific installations instructions provided with each component's documentation, as the installation requirements for the components within Everyplace Server may vary. The following order of installation is recommended in order to ensure key components are installed and working.

1. Install SecureWay Directory
2. Install Tivoli Personalized Services Manager
3. Install other Everyplace Server components

Installation Tip	At any time during installation, you can click the Back button to return to previous panels to review or change the information. You have the opportunity to review all entries on the Final Selection Confirmation panel before the installation actually begins.
	If you cancel Everyplace Server installation prior to installing any of the Everyplace Suite components, you must use `SMIT`, `SMITTY`, or `admintool` to remove the Everyplace Server files.

Step one: Install SecureWay Directory

Everyplace Server integrates its components over a supporting environment. This supporting environment enables information sharing and cross-component communications. The key parts of this supporting environment are LDAP directory services (SecureWay Directory), DB2 Universal Database, and IBM HTTP Server. This protocol provides access to the X.500 directory over a TCP or SSL connection. LDAP lets you store information in a directory service and query it in a database fashion.

SecureWay Tip	Everyplace Server relies on a specific directory schema that is only implemented in SecureWay Directory. Therefore, SecureWay Directory is a prerequisite for all Everyplace Server components. It is strongly recommended that SecureWay Directory be deployed within any Everyplace Server domain.

Important information is frequently shared by Everyplace Server components. Enabling such information sharing is part of the effort to achieve suite integration. To achieve this, the Everyplace Server architecture is based on a common and shared directory structure as well as a dynamically shared database structure. Information about systems, users, devices, network, and configurations are stored in the shared directory structure, while more dynamic information (such as session information) is stored in the dynamic database structure. Within Everyplace Server, the directory structure is build on the SecureWay Directory LDAP server and the dynamic information sharing is built on DB2 Universal Database.

Note: Everyplace Server installation is not supported in an NIS (Network Information Service) environment for either Solaris or AIX.

Install SecureWay Directory first	It is important to install SecureWay Directory first and confirm that it is configured and running correctly. Many other Everyplace Server components require that SecureWay Directory be installed and running before they can be installed.

To install SecureWay Directory:

1. Begin Setup Manager and follow the prompts. Start Setup Manager by running `install.sh`. For detailed information on Setup Manager and starting the installation see Prepare for installation.
2. Setup Manager requires Java 1.3 to run. The installation program automatically attempts to locate it. If it cannot be found, you are prompted to install or locate Java 1.3 on the local machine.

3. Review the following documentation before proceeding:
   - InfoCenter: Contains Everyplace Server introduction, planning migration, installation, configuration, administration, and uninstallation information. Also contains information on the other Everyplace Server components.
   - Readme: Contains Everyplace Server late-breaking news and known issues.
4. Enter your license key when asked. For additional information on product keys or installation scenarios, see Install Strategies.
5. Specify an XML file to use in subsequent installations. See XML Automated Installation for more information on this step.
6. Select **Install SecureWay Directory on this server** (Option 1) on the "Information Sharing Options" panel.
   - This decision is a critical step in the installation process. Be sure to consider the deployment situation.
   - Directory information entered when setting up SecureWay Directory is required later during the installation process.
   - The default LDIF filename and path is `/tmp/everyplace/everyplace`.
   - The following table illustrates how the options affect installation. See Related information for more post installation and external information.

Sharing Options Table	**Option 1: Local**	**Option 2: Remote**	**Option 3: LDIF**
	Setting up SecureWay Directory, LDAP, and installing components	**Retrieve information from LDAP**	**Share information via File System**
Install SecureWay Directory	Yes - installs on the local machine.	No - retrieves the information from the existing SecureWay Directory (LDAP) server in the Everyplace domain.	Yes - installs without LDAP. Configuration information imported later. SecureWay Directory can be selected for install.
LDAP Access User ID	Yes.	Yes, the active SecureWay Directory is required.	Yes.
Import LDIF file	Done automatically.	No need.	Done automatically.

Usage	Recommended first install in the domain.	Recommended subsequent install.	- Sophisticated user. - Components installed before SecureWay Directory. - When LAN is not available. - When migrating from earlier version.
Required Data	- Directory suffix - Object type - Directory database name, instance, and home directory - Directory host port - Group - User ID - Password	- LDAP host name - User ID - Password	LDIF filename
Comments	- Be sure to include `dc=` before each part of the directory suffix. - Be sure to include `cn=` before the user ID value.	Be sure to include `cn=` before the user ID value.	- Use the same LDIF on subsequent installations within domain. - LDIF file cannot be shared between AIX and Solaris systems. - LDIF configuration information is automatically imported.

Importing LDIF Tip	If there is a need to import the LDIF manually, use the `ldif2db` utility. The following command must be run on the SecureWay Directory Server: `ldif2db -i file_name` where `file_name` is the full path of the LDIF file that was specified during installation.

7. Do not select any additional components from the "Everyplace Server Components" panel. DB2 Universal Database and IBM HTTP Server are required to use this component and are installed silently.

8. Click **Install** on the "Installation Summary" panel.
9. You are prompted for the correct CDs.
10. Verify SecureWay Directory
    - From a command prompt:
      . Run the following command: `dmt`
      b. Select **Rebind**
      c. Select **Authenticated**
      d. Enter user ID and Password
      e. Select **Browse Tree**
      f. Check under the SPD suffix for device and network profiles.
    - From a browser:
      . Start the "Administration Console" from a browser
      b. Go to: http://<hostname>/ldap
      c. Enter UserId and Password
      d. Check the status of the server by selecting: **Current State > Server status**
11. See Related information for post installation and external information links.

Step two: Install Tivoli Personalized Services Manager

Second, install Tivoli Personalized Services Manager on a different machine than the one SecureWay Directory was installed on.

Oracle Tips	Be sure to install the Tivoli Oracle integration package and the Oracle8i software as outlined in "Installation of Oracle Database software" in the Install Oracle section before installing Everyplace Server.

To install Tivoli Personalized Services Manager:

1. Begin Setup Manager and follow the prompts. Start Setup Manager by running `install.sh`. For more information on Setup Manager and starting the installation see Prepare for installation.
2. Enter your license key when asked. For additional information on product keys or installation scenarios, see Install Strategies.
3. Select **Retrieve existing Everyplace Server information from a SecureWay Directory server** (Option 2) on the "Installation and Configuration Sharing Options" panel. For additional information on working with SecureWay Directory sharing options, see the Sharing Options Table.

   . Enter the SecureWay Directory password on the "Existing SecureWay Directory Service Information" panel.

   b. If you install Tivoli Personalized Services Manager using the LDIF file, you cannot install the component using a remote DB2 database server since SecureWay Directory is not available for the Everyplace Server installation program. You must either use a local DB2 database server or a local or remote Oracle database server. Therefore, do not use the Share Information via File System (LDIF) option unless you are installing Tivoli Personalized Services Manager on a local DB2 database server.

4. Select **Tivoli Personalized Services Manager** on the "Everyplace Server components" panel. Do not select any additional components. WebSphere Application Server is required to use this component and is installed silently. Some manual steps are necessary to configure Tivoli Personalized Services Manager with WebSphere Application Server.

Remote DB2 Tip	If you install Tivoli Personalized Services Manager or Everyplace Wireless Gateway using a remote DB2 server, you must determine whether the `db2inst1` user ID exists on the local machine before running the Everyplace Server installation program. For AIX: Enter `SMIT` to check the user ID. For Solaris: Enter `admintool` to check the user ID. If the `db2inst1` user ID exists on the local machine, but the `/home/db2inst1` directory does NOT exist on the local machine, you must erase the `db2inst1` user ID, using `SMIT` for AIX or `admintool` for Solaris.

5. A database management system is required to use this component. You are prompted with a choice of either DB2 Universal Database or Oracle.
   - If you want to use Oracle, it must be installed before continuing. If Oracle is not already installed, cancel the installation, install Oracle, and start the installation again. See Using Oracle for more information.
   - If you want to use DB2 Universal Database, decide if you want to use a Local or Remote database. If you select Local and do not have DB2 installed, Setup Manager installs it for you.
6. Click **Install** on the "Installation Summary" panel.
7. You are prompted for the correct CDs.

Step three: Install other Everyplace Server components

Installation of the remaining Everyplace Server may take place in any order and on any machine. All Everyplace Server components are displayed in the panel. Setup Manager verifies that prerequisite and corequisite software are installed before beginning. A message is displayed if prerequisites are missing. Some required components are silently installed when necessary.

Upgrade Tip	It may be necessary to upgrade component or prerequisite software to complete the installation. Follow the instructions on the installation panels to verify and upgrade as necessary. It may sometimes be necessary to downgrade to an earlier version of the software to ensure that Everyplace Server runs properly. Without the recommended versions of the prerequisite and corequisite software, Everyplace Server may not work as expected.

For this installation scenario, a third computer system houses additional components.

1. Begin Setup Manager and follow the prompts. Start Setup Manager by running `install.sh`. For more information on Setup Manager and starting the installation see Prepare for installation.
2. Enter your license key when asked. For additional information on product keys or installation scenarios, see Install Strategies.
3. Specify an XML file to use in subsequent installations. See XML Automated Installation for more information on this step.
4. Select **Retrieve existing Everyplace Server information from a SecureWay Directory server** (Option 2) on the "Installation and Configuration Sharing Options" panel. For additional information on working with SecureWay Directory sharing options, see the Sharing Options Table.
   . Enter the SecureWay Directory password on the "Existing SecureWay Directory Service Information" panel.
5. Select the components you wish to install from the left side of the "Everyplace Server Components" panel. For example, if you desired to install authentication components, you might select Edge Server Caching Proxy, Edge Server Load Balancer, WebSEAL-Lite, Everyplace Active Session Table, and Tivoli SecureWay Policy Director (optional). For additional information on the components and their role in Everyplace Server, see Product Components and Install Strategies.
6. Setup Manager prompts you for additional information, such as Database Management System details, based on your install selections.
7. Many components require configuration at various points before any files are copied to the system. If you select a component that requires pre-installation configuration, a "Component Configuration" panel appears and prompts you for information specific to your environment.

Configuration Tip	If there are other instances of components that you are currently installing in the Everyplace Server, you are given configuration options for each component with other instances already in the domain. Each instance is listed by server name. You can select one of the instances to configure the current component with the same parameters, or you can select **New configuration** if you don't want to use any of the pre-installed configuration parameters.

8. Setup Manager prompts you to enter appropriate user ID information. You can select either single or multiple user IDs for component administration. Some of the Everyplace Server components require a user ID, group, and password for configuration. You have the option of configuring these components with the same ID or different IDs.
    - The single user ID option prompts you for the user information once for all components.
    - The different user ID option prompts you for user information once for each component.
    - Do not include `cn=` as part of the user ID values here. You generally enter `cn=` for user ID fields associated with SecureWay Directory only.
9. Choose **Install** from the "Installation Summary" panel to confirm. All selected components, subcomponents, prerequisite components, and configuration information are displayed. You can click the **Back** button to go back to the component selection panel to add or remove components from the list or to change configuration information. When the **Install** button is clicked, the installation begins.

Related information

- Tivoli Personalized Services Manager post installation configuration
- WebSphere Everyplace Server authentication using WebSEAL-Lite
- Configure Everyplace Wireless Gateway
- Configure User preferences
- External information

Enable SSL on IBM HTTP Server for Tivoli Personalized Services Manager

If Tivoli Personalized Services Manager is installed, you should create a Secure Sockets Layer (SSL) certificate so that IBM HTTP Server can be started for the first time. The self-signed certificate created here is for temporary use to verify the success of your installation. After installation, you may use your own certificate. The manual steps that follow need to be completed after Tivoli Personalized Services Manager components are installed.

The following procedure creates a key file to enable SSL for IBM HTTP Server.

1. Run the following commands:
   - `xhost + localhost` on your machine terminal
   - `export DISPLAY=MachineX:0.0` where MachineX is the name of your machine
   - `ikeyman`

   Note: ikeyman launches the GUI. On IBM AIX machines, you may need to set JAVA_HOME and PATH directories. You can also Telnet to the server or client machine.
2. Select **Key Database File**
3. Select **New**
4. In the "New" window, designate the key database type to **CMS key database file**
5. Accept the default for the key database name: `key.kdb`
6. Assign the location as `/`

   Note: If you store the key database in a location other than the root directory or use a different filename, update the line `Keyfile /key.kdb` with the new name and location. The line is located in the /usr/HTTPServer/conf/httpd.conf or `/opt/IBMHTTPD/conf/httpd.conf` file.
7. Click **OK**
8. In the "Password Prompt" window, enter and confirm your password
9. Select the following to save your password to the file: **Stash the password to a file?**
10. Click **OK**
11. In the "Key Database" content frame, select **Personal Certificates**
12. Click **New Self-Signed**
13. Complete the "Create New Self-Signed Certificate" window
14. Click **OK**

Install and configure Tivoli Personalized Serviced Manager servlets on WebSphere Application Server

This section has instructions to guide you through manual installation of Tivoli Personalized Service Manager servlets on WebSphere Application Server. This procedure relies on Setup Manager to install most of the required files on the system, edit property files, and update IBM HTTP Server configurations. You should have followed the instructions in Installation Steps, on how to install Tivoli Personalized Services Manager, and in Enable SSL before proceeding.

AIX versus Sun Solaris	These instructions are written using AIX directories and filenames. The instructions also apply to Sun Solaris when the following substitutions are made.
	● You must substitute the appropriate operating system root for Solaris.
	AIX = `/usr`
	Solaris = `/opt`
	● IBM HTTP Server exists in a different path when installed on Solaris and must be substituted.
	AIX = `/usr/HTTPServer`
	Solaris = `/opt/IBMHTTPD`
	● For Solaris, command line references to AIX should be dropped in `createDMS*.xml` and `createDMS*.ksh` filenames. For example, the execute command would change as follows for Solaris:
	AIX = `chmod 755 createDMScare_AppServerAIX.ksh`
	Solaris = `chmod 755 createDMScare_AppServerSUN.ksh`
	● The convention for DMS Xml and .ksh files are different. Remove `AIX` from the filenames. For example:

	AIX =
	`createDMS_WebAppAIX.ksh`
	Solaris =
	`createDMS_WebApp.ksh`

Install Tivoli Personalized Services Manager servlets

1. Comment out the following ApJServMount entries in the httpd.conf file found in /usr/HTTPServer/conf.
   - `ApJServMount /dmscare /dmscare`
   - `ApJServMount /authentication`
   - `ApJServMount /dmserver /dmserver`
   - `ApJServMount /perso`
   - `include /usr/jakarta-tomcat-3.2.1/conf/tomcat-apache.conf`

   Note: Additional Tivoli Personalized Services Manager component installs may make new entries in the httpd.conf file. After each install, check to ensure that all of the above entries are commented out.
2. Edit createSMdefault_host.xml, located in /usr/TivTSM/install/etc as follows:
   - Replace `000.000.000.000` with `<yourIPaddress>`
   - Replace `fully_qualified_hostname with <yourfully_qualifed_hostname>`
   - Replace `hostname` with `<yourhostname>`
3. Clean up extraneous files.
   - If running on AIX, remove all Solaris related files by typing rm *SUN.*
   - If running on Solaris, remove all AIX related files by typing rm *AIX.*

Configure default server

1. Before starting the WebSphere Application Server, you must source your database path. Add the following to the top of: /usr/IBMWebAS/bin/startupServer.sh:

   DB2 on AIX or Solaris

   ```
   . /home/db2inst1/sqllib/db2profile
   TSM_DB_CLASSPATH=/home/db2inst1/sqllib/java12/db2java.zip
   export TSM_DB_CLASSPATH
   ```

   Oracle on AIX or Solaris

   ```
   #!/bin/sh
   dir=`csh -fc "echo ~oracle8"`
   . $dir/.profile
   TSM_DB_CLASSPATH=/db/app/oracle/products/8.1.7/OraHome1/jdbc/lib/classes12.zip
   export TSM_DB_CLASSPATH
   ```

2. Start WebSphere Application Server. See step 3 in the Start and Stop Components section below for instructions.
3. Enter `cd /usr/TivTSM/install/etc`
4. Create the WebSphere Application Server for the default host:
   - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMdefault_host.xml -adminNodeName <yournodename>`
5. Start the WebSphere Application Server Administration Console by performing the following commands:
   - `cd /usr/IBMWebAS/bin`
   - `./adminclient.sh &`
6. Expand the <nodename> tree until you see the default host.
7. Highlight the default host by left clicking on it.

8. Under the **General Tab**, the following information edited during installation should be visible:
   - aliases with IP
   - fully qualified hostname
   - hostname
   - ports 80, 443, 8080, and 12080

Start and Stop Components

1. To start IBM HTTP Server.
   - `cd /usr/HTTPServer/bin`
   - `./apachectl start`

   Note: On AIX, you may not be able to restart IBM HTTP Server without rebooting the machine. See Operating system support and requirements in Requirements and prerequisites for information on how to fix this problem with an APAR.
2. To stop IBM HTTP Server:
   - `CD /usr/HTTPServer/bin`
   - `./apachectl stop`
3. To start WebSphere Application Server:
   - `CD /usr/IBMWebAS/bin`
   - if starting for the first time: `./setupCmdLine.sh`
   - `./startupServer.sh &`
   - `tail -f ../logs/tracefile`
   - The log says "open for e-business"when the WebSphere Application Server has started.
   - CTRL-C stops the tail.
   - log files are in /usr/IBMWebAS/logs
4. To stop WebSphere Application Server from the Administrative Console:
   - Right click the host node
   - Select **stop**
5. To start the WebSphere Application Server Administrative Console:
   - `CD /usr/IBMWebAS/bin`
   - `./adminclient.sh &`
6. To start a WebSphere Application Server servlet with the administration console:
   - Right click on the WebSphere Application Server servlet you wish to start
   - Select **Start**

   Note: You must start the WebSphere Application Server for each web application after creating.

Install web applications

There may be as many as eight web applications installed during the Tivoli Personalized Services Manager install. The steps for installing and validating each are similar.

Enrollment

Required WAR files are in `/usr/TivTSM/enroll`

1. Edit createSMEnroll_AppServerAIX.xml
   - Replace `hostname` with `<yourhostname>`
2. Edit `createSM_HSMEnrollAIX.ksh`
   - Replace hostname with `<yourhostname>`
3. Start WebSphere Application Server by performing the following commands:

    - `CD /usr/IBMWebAS/bin`
    - if starting for the first time: `./setupCmdLine.sh`
    - `./startupServer.sh &`
    - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
    - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMEnroll_AppServerAIX.xml -adminNodeName <yournodename>`
    - `chmod 755 ./createSM_HSMEnrollAIX.ksh`
6. Convert the WAR file to create the needed directories and files
    - `CD /usr/TivTSM/install/etc`
    - `./createSM_HSMEnrollAIX.ksh`
7. The enroll.html is in /usr/HTTPServer/htdocs/tsm and has been updated with the appropriate hostname by Tivoli Personalized Services Manager
    - If you have not enabled SSL, you must edit enroll.html to work with port 443. See Enable SSL for more information.
8. Copy intro.html from /usr/jakarta-tomcat-3.2.1/webapps/HSMEnroll/ to /usr/TivTSM/HSMEnroll/web/
9. Start the Administration Console. See previous section on Start and Stop Components
10. In the Administration Console, expand the tree structure to see **SMenroll_AppServer** > **SMenrollServletEngine** > **HSMEnroll**
11. Expand **HSMEnroll** so that you see jsp11
12. Highlight **jsp11** and right click
13. Select **Remove** from the pop up menu
14. Click the icon that looks like a wand for Wizards
15. Click **Add a JSP Enabler**
16. Select **Create JSP 1.0 Enabler**
17. The Web Application is HSMEnroll
18. Start SMHSMEnroll_Appserver by highlighting **SMenroll_Appserver** and clicking the green **>** button
19. Start IBM HTTP Server. If it was running during install, you should stop and restart it
20. To validate, point your browser to http://<yourhostname>/tsm/enroll.html

Selfcare

Required WAR files are in `/usr/TivTSM/selfcare`.

1. Edit createSMSelfcare_AppServerAIX.xml
    - Replace `hostname` with <`yourhostname`>
2. Edit `createSM_HSMSelfcareAIX.ksh`
    - Replace hostname with <`yourhostname`>
3. Start WebSphere Application Server by performing the following commands:
    - `CD /usr/IBMWebAS/bin`
    - if starting for the first time: `./setupCmdLine.sh`
    - `./startupServer.sh &`
    - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
    - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMSelfcare_AppServerAIX.xml -adminNodeName <yournodename>`

    - `chmod 755 ./createSM_HSMSelfcareAIX.ksh`
6. Convert the WAR file to create the needed directories and files
    - `CD /usr/TivTSM/install/etc`
    - `./createSM_HSMSelfcareAIX.ksh`
7. Copy the following files from /usr/jakarta-tomcat-3.2.1/webapps/HSMSelfcare/WEB-INF/classes to /usr/TivTSM/HSMSelfcare/servlets/
    - selfcare.properties
    - DefaultAuthenticator.properties
    - DBPool.properties - be sure the dbConnect line is not commented out.
8. Copy menu.jsp from /usr/jakarta-tomcat-3.2.1/webapps/HSMSelfcare/jsp/ to /usr/TivTSM/HSMSelfcare/web/jsp
9. Start the Administration Console. See previous section on Start and Stop Components
10. In the Administration Console, expand the tree structure to see **SMselfcare_AppServer** > **SMselfcareServletEngine** > **HSMSelfcare**
11. Expand **HSMSelfcare** so that you see jsp11
12. Highlight **jsp11** and right click
13. Select **Remove** from the pop up menu
14. Click the icon that looks like a wand for Wizards
15. Click **Add a JSP Enabler**
16. Select **Create JSP 1.0 Enabler**
17. The Web Application is HSMSelfcare
18. Start SMHSMSelfcare_Appserver by highlighting **SMselfcare_Appserver** and clicking the green **>** button
19. Start IBM HTTP Server. If it was running during install, you should stop and restart it
20. To validate, point your browser to http://<yourhostname>/tsm/selfcare.html

Customer care

Required WAR files are in `/usr/TivTSM/custcare`.

1. Edit createSMcustcare_AppServerAIX.xml
    - Replace `hostname` with <`yourhostname`>
2. Edit `createSM_HSMcustcareAIX.ksh`
    - Replace hostname with <`yourhostname`>
3. Start WebSphere Application Server by performing the following commands:
    - `CD /usr/IBMWebAS/bin`
    - if starting for the first time: `./setupCmdLine.sh`
    - `./startupServer.sh &`
    - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
    - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMcustcare_AppServerAIX.xml -adminNodeName <yournodename>`
    - `chmod 755 ./createSM_HSMcustcareAIX.ksh`
6. Convert the WAR file to create the needed directories and files
    - `CD /usr/TivTSM/install/etc`
    - `./createSM_HSMcustcareAIX.ksh`
7. Start the Administration Console. See previous section on Start and Stop Components
8. In the Administration Console, expand the tree structure to see **SMcustcare_AppServer** > **SMcustcareServletEngine** > **HSMcustcare**

9. Expand **HSMcustcare** so that you see jsp11
10. Highlight **jsp11** and right click
11. Select **Remove** from the pop up menu
12. Click the icon that looks like a wand for Wizards
13. Click **Add a JSP Enabler**
14. Select **Create JSP 1.0 Enabler**
15. The Web Application is HSMcustcare
16. Start SMcustcare_Appserver by highlighting **SMcustcare_Appserver** and clicking the green **>** button
17. Start IBM HTTP Server. If it was running during install, you should stop and restart it
18. To validate, point your browser to http://<yourhostname>/tsm/custcare.html

Authentication

Required WAR files are in `/usr/TivTSM/authentication`.

1. Edit createSMauthentication_AppServerAIX.xml
    - Replace `hostname` with `<yourhostname>`
2. Edit `createSM_authenticationAIX.ksh`
    - Replace hostname with `<yourhostname>`
3. Start WebSphere Application Server by performing the following commands:
    - `CD /usr/IBMWebAS/bin`
    - if starting for the first time: `./setupCmdLine.sh`
    - `./startupServer.sh &`
    - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
    - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMauthentication_AppServerAIX.xml -adminNodeName <yournodename>`
    - `chmod 755 ./createSM_authenticationAIX.ksh`
6. Convert the WAR file to create the needed directories and files
    - `CD /usr/TivTSM/install/etc`
    - `./createSM_authenticationAIX.ksh`
7. Copy the following files from /usr/jakarta-tomcat-3.2.1/webapps/authentication/WEB-INF/classes to /usr/TivTSM/authentication/servlets/
    - DefaultAuthenticator.properties
    - DBPool.properties
8. Start the Administration Console. See previous section on Start and Stop Components
9. In the Administration Console, expand the tree structure to see **SMauthentication_AppServer** > **SMauthenticationServletEngine** > **authentication**
10. Expand **authentication** so that you see jsp11
11. Highlight **jsp11** and right click
12. Select **Remove** from the pop up menu
13. Click the icon that looks like a wand for Wizards
14. Click **Add a JSP Enabler**
15. Select **Create JSP 1.0 Enabler**
16. The Web Application is authentication

17. Start SMauthentication_Appserver by highlighting **SMauthentication_Appserver** and clicking the green **>** button
18. Start IBM HTTP Server. If it was running during install, you should stop and restart it
19. To validate, point your browser to http://<yourhostname>/authentication/server

SDPServlet

Required servlets are in `/usr/TivTSM/sdp`.

1. Edit createSMsdpservlet_AppServerAIX.xml
   - Replace `hostname` with `<yourhostname>`
2. Edit `createSM_HSMsdpservletAIX.ksh`
   - Replace hostname with `<yourhostname>`
3. Start WebSphere Application Server by performing the following commands:
   - `CD /usr/IBMWebAS/bin`
   - if starting for the first time: `./setupCmdLine.sh`
   - `./startupServer.sh &`
   - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
   - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMsdpservlet_AppServerAIX.xml -adminNodeName <yournodename>`
   - `chmod 755 ./createSM_HSMsdpservletAIX.ksh`
6. Copy the HSMsdpservlet.war file to a different location and rename it to sdp.war using the following command:
   - `mv /usr/TivTSM/sdp/HSMsdpservlet.war /usr/TivTSM/sdp.war`
7. Remove the existing /usr/TivTSM/sdp directory
   - `rm -r /usr/TivTSM/sdp`
8. Edit `createSM_HSMsdpservletAIX.ksh` so that
   - `WEBAPP_PATH=sdp`
   - `WEBAPP_NAME=sdp`
   - `WAR_FILENAME=/usr/TivTSM/sdp.war`
9. Convert the WAR file to create the needed directories and files
   - `CD /usr/TivTSM/install/etc`
   - `./createSM_HSMsdpservletAIX.ksh`
10. Copy the HSMsdpservlet.jar file from /usr/TivTSM/sdp/servlets to /usr/TivTSM/sdp
    - `cp /usr/TivTSM/sdp/servlets/HSMsdpservlet.jar /usr/TivTSM/sdp`
11. Start the Administration Console. See previous section on Start and Stop Components
12. In the Administration Console, expand the tree structure to see **SMsdp_AppServer** > **SMsdpServletEngine** > **sdp**
13. Left-click on **sdp**, then click on the **Advanced** tab in the right pane of the Administration Console.
    - Change the **CLASSPATH** to /usr/TivTSM/sdp
    - Click on **Apply** when you've completed the change.
14. Expand **sdp** so that you see sdpservlet
15. Left-click on **sdpservlet** and click on the **General** tab in the right pane of the Administration Console.
16. Highlight the top entry in the "Servlet Web Path List" box and click the **Edit** button.
17. In the dialogue box that appears, change the "Servlet Path" entry that appears from /sdp/servlets/sdpservlet to /sdp/sdpservlet. Click the **OK** button.

18. Click the **Apply** button on the Administration Console.
19. Expand **sdp** so that you see jsp11
20. Highlight **jsp11** and right click
21. Select **Remove** from the pop up menu
22. Click the icon that looks like a wand for Wizards
23. Click **Add JSP Enabler**
24. Select **Create JSP 1.0 Enabler**
25. The Web Application is sdp
26. Add the following line to /usr/HTTPServer/conf/httpd.conf:
    - `Listen 8080`
27. Start SMsdp_Appserver by highlighting **SMsdpservlet_Appserver** and clicking the green **>** button
28. Start IBM HTTP Server. If it was running during install, you should stop and restart it
29. To validate, point your browser to http://<yourhostname>:8080/sdp/sdpservlet

HSMConsole and DMConsole

Required WAR files are in /usr/jakarta-tomcat-3.2.1/webapps.

1. Edit createSMConsole_AppServerAIX.xml
    - Replace `hostname` with <`yourhostname`>
2. Edit `createSM_HSMConsoleAIX.ksh`
    - Replace hostname with <`yourhostname`>
3. Start WebSphere Application Server by performing the following commands:
    - `CD /usr/IBMWebAS/bin`
    - if starting for the first time: `./setupCmdLine.sh`
    - `./startupServer.sh &`
    - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
    - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMConsole_AppServerAIX.xml -adminNodeName <yournodename>`
    - `chmod 755 ./createSM_HSMConsoleAIX.ksh`
6. Convert the WAR file to create the needed directories and files
    - `CD /usr/TivTSM/install/etc`
    - `./createSM_HSMConsoleAIX.ksh`
7. Start the Administration Console. See previous section on Start and Stop Components
8. In the Administration Console, expand the tree structure to see **SMConsole_AppServer** > **SMConsoleServletEngine** > **<HSMConsole>**
9. Expand **<HSMConsole>** so that you see jsp11
10. Highlight **jsp11** and right click
11. Select **Remove** from the pop up menu
12. Click the icon that looks like a wand for Wizards
13. Click **Add a JSP Enabler**
14. Select **Create JSP 1.0 Enabler**
15. The Web Application is HSMConsole
16. Start SMConsole_Appserver by highlighting **SMConsole_Appserver** and clicking the green **>** button
17. Start IBM HTTP Server. If it was running during install, you should stop and restart it

18. To validate, point your browser to `http://<yourhostname>/HSMConsole/servlet/ConsoleServlet` for AIX orhttp://<yourhostname>/HSMConsole/servlet/ConsoleServlet for Solaris. You can also go to http://<yourhostname> and click on **tsm**, then **console** to download both consoles.

perso

Required WAR files are in /usr/TivTSM/personal.

1. Edit createSMperso_AppServerAIX.xml
   - Replace `hostname` with `<yourhostname>`
2. Edit `createSM_persoAIX.ksh`
   - Replace hostname with `<yourhostname>`
3. Start WebSphere Application Server by performing the following commands:
   - `CD /usr/IBMWebAS/bin`
   - if starting for the first time: `./setupCmdLine.sh`
   - `./startupServer.sh &`
   - log files are in /usr/IBMWebAS/logs
4. Enter `CD /usr/TivTSM/install/etc`
5. Create the WebSphere Application Server for the web application:
   - `/usr/IBMWebAS/bin/XMLConfig.sh -import createSMperso_AppServerAIX.xml -adminNodeName <yournodename>`
   - `chmod 755 ./createSM_persoAIX.ksh`
6. Convert the WAR file to create the needed directories and files
   - `CD /usr/TivTSM/install/etc`
   - `./createSM_persoAIX.ksh`
7. Copy the following files from /usr/jakarta-tomcat-3.2.1/webapps/perso/WEB-INF/classes to /usr/TivTSM/perso/servlets/
   - DefaultAuthenticator.properties
   - DBPool.properties
8. Start the Administration Console. See previous section on Start and Stop Components
9. In the Administration Console, expand the tree structure to see **SMperso_AppServer** > **SMpersoServletEngine** > **perso**
10. Remove the jsp11 servlet from under:

```
SMperso_AppServer
SMpersoServletEngine
perso
```

11. Click the icon that looks like a wand for Wizards
12. Click **Add a JSP Enabler**
13. Select **Create JSP 1.0 Enabler**
14. The Web Application is perso
15. Start SMperso_Appserver by highlighting **SMperso_Appserver** and clicking the green **>** button
16. Start IBM HTTP Server. If it was running during install, you should stop and restart it
17. To validate, point your browser to http://<yourhostname>/perso/home

Install DMS Xcare

If you are planning to use SSL, configure IBM HTTP Server to use it.

The `dmscare.war` is in /usr/TivTSM/dsmcare by the installer.

1. Check to ensure that the createDMScare_AppServerAIX.xml and createDMScare_WebAppAIX.ksh files have been properly configured to reflect the correct hostname.

- createDMScare_AppServerAIX.xml: The value of the node name must be set to the DMS server hostname.

  ```
  <websphere-sa-config>
  <node name="hostname" action="update">
  <deployed-jar-directory>$server_root$/deployedEJBs</deployed-jar-directory>
  <dependent-classpath></dependent-classpath>
  ```

- createDMScare_WebAppAIX.ksh: The value of the node entity must be set to the DMS server hostname

  ```
  WAR_FILENAME=/usr/TivDMS/dmscare.war
  WEBAPP_DEST=/usr/TivDMS
  ADMIN_NODE=hostname
  NODE=hostname
  ```

2. From /usr/TivTSM/install/etc, type:
   - `/usr/IBMWebAS/bin/XMLConfig.sh -import createDMScare_AppServerAIX.xml -adminNodeName <yournodename>`
3. From /usr/TivTSM/install/etc, type:
   - `chmod 755 createDMScare_WebAppAIX.ksh`
   - `./createDMScare_WebApp.ksh`
4. Customize Xcare with `SelfCare.properties` and `CustomerCare.properties` in /usr/TivDMS/dmscare/servlets. Configuration options are found in the DMS Developer's Guide.
5. After restarting WebSphere Application Server and IBM HTTP Server, Xcare should be ready for use after you start the DMScare_AppServer.
6. To verify Xcare, either selfcare or custcare should be running. Login with a valid user name and password.
   - Selfcare: Click on the **devices** button to see a list of devices and some options. If no devices are present, you will see a note indicating that.
   - Custcare:

     . Search on a business or individual.

     b. Select a name from the search list.

     c. Click devices to see the same results as described for selfcare.

Install DMServer

The `dmserver.war` is in /usr/TivDMS by the installer.

1. Check to ensure that the createDMS_AppServerAIX.xml and createDMS_WebAppAIX.ksh files have been properly configured to reflect the correct hostname.
   - createDMS_AppServerAIX.xml: The value of the node name must be set to the DMS server hostname.

     ```
     <websphere-sa-config>
     <node name="hostname" action="update">
     <deployed-jar-directory>$server_root$/deployedEJBs</deployed-jar-directory>
     <dependent-classpath></dependent-classpath>
     ```

   - createDMS_WebAppAIX.ksh: The value of the node entity must be set to the DMS server hostname

     ```
     WAR_FILENAME=/usr/TivDMS/dmserver.war
     WEBAPP_DEST=/usr/TivDMS
     ADMIN_NODE=hostname
     NODE=hostname
     ```

2. From /usr/TivTSM/install/etc, type:
   - `/usr/IBMWebAS/bin/XMLConfig.sh -import createDMS_AppServerAIX.xml -adminNodeName <yournodename>`
3. From /usr/TivTSM/install/etc, type:
   - `chmod 755 createDMS_WebAppAIX.ksh`
   - `./createDMS_WebAppAIX.ksh`

4. Copy and modify as needed Transaction.properties and ConsoleTransaction.properties from /usr/TivDMS/dmserver/WEB-INF/classes to /usr/TivDMS/dmserver/servlets. If necessary, create the servlet directory in /usr/TivDMS/dmserver first.
5. After restarting WebSphere Application Server, the Administration Console, and IBM HTTP Server, Xcare should be ready for use after you start the DMScare_AppServer.
6. To check the DMServer, you must first download and install the HSMConsole. Refer to HSMConsole for instructions.
   - From the Start Menu, go to **Programs > Tivoli > Device Manager**
   - Log in with a valid user name and password, also supplying the name of the machine on which the DMServer is running.
   - A successful login with valid menus indicate that the DMServer is providing the correct information to the console.

Clean up

After validating the servlets you have just moved, as described above, clean up materials that are no longer in use by typing the following at the command prompt:

- `CD /`
- For AIX: `rm -rf /usr/jakarta-tomcat-3.2.1`
- For Solaris: `rm -rf /opt/jakarta-tomcat-3.2.1`

This frees up approximately 30MB of disk space.

Related information

- Installation Steps
- Enable SSL on IBM HTTP Server for Tivoli Personalized Services Manager
- Uninstalling the WebSphere Everyplace Server components

Troubleshooting installation

This section contains possible error messages and workarounds that may be appear when installing and configuring Everyplace Server components.

- Errors messages
- Setup Manager windows not appearing
- DB2 installation errors
- Logging

Errors messages

If errors occur during installation, the program notifies you regarding the specific problem. Follow the instructions indicated by the error message. Some problems may require termination of the installation program.

Setup Manager windows not appearing

- **Installation panel disappears on Solaris systems:** If you click in the background area outside the install window, the install window goes to the background and is not visible. To bring the install window back in view, press the **Alt-Tab** keys simultaneously.
- **Setup Manager looks like it has stopped responding, no action can be done:** Setup Manager includes many dialog boxes that require answers to questions. This problem could be caused by a dialog box that opened behind another window and is waiting for an answer. If this is true, locate the dialog box and answer the question to continue with installation. Possible ways to get around this problem include:
  - Do not use the "focus follows pointer" and "autoraise" features in your window manager. This will cause some of the smaller dialogs to be obscured by the larger ones should the pointer be moved inadvertently.
  - Be aware where you are clicking to avoid inadvertently obscuring the smaller dialogs.
  - Should a dialog become obscured, try rearranging the windows currently displayed. Failing that, Alt-Tab will cycle through all parent (major) displayed windows which may reveal an obscured dialog box. Depending on the user's window manager, Ctrl-Tab may cycle through child (minor) windows which would allow the user to find the obscured window.
  - Use an alternate window manager, such as twm, instead of CDE's built-in window manager.

  Note: This problem has been observed on AIX machines running in a Japanese locale, however this problem may occur in other environments.

- **Install Summary and Status panel not showing selected components:** To correct this problem, resize the panel. This causes the panel to refresh and update the list of components.
- **Insert Next CD dialog not appearing:** The Insert Next CD dialog may not appear on slow X Window servers or if you have a slow connection to an X Windows server.

DB2 installation errors

During the installation of DB2, DB2 attempts to install NetQuestion SBCS Search Engine 1.2.3.1. This produces errors that are logged in `/tmp/db2udbee.log`. On AIX the component is identified as IMNSearch.rte.SBCS, while on Solaris, it is identified as IMNSBCS.

Everyplace Server does not use this package and the errors can be ignored. However, AIX users will see the following in SMIT:

```
Fileset              Level    State   Descriptions
------------------   -------  -----   ------------
IMNSearch.rte.SBCS   1.2.3.1    ?     NetQuestion SBCS Search Engine
```

The State codes are as follows:
A is Applied
B is Broken
C is Committed
O is Obsolete (partially migrated to newer version)
? is Inconsistent State ...Run lppchk -v

Logging

During installation, all actions and outcomes are logged to an installation log file in `/tmp/everyplace_install.log`. This file contains a sequence of information that can assist you in identifying and analyzing problems. This file is cumulative over multiple Everyplace Server installations.

Details about Setup Manager execution are logged to a trace log file in `/tmp/everyplace_install.trace.number`. A new trace file is created each time you run Setup Manager. If a trace file already exists in the directory Setup Manager appends a sequential number to the end of the filename. The most recent trace file does not have a number appended to the end of it.

These files are in ASCII text format and can be viewed using any text editor.

Migration scenarios

There is a standard migration period which lasts from the time migration begins until Everyplace Server is online. Depending on the complexity of the customer environment, this migration period could be extensive. To assist with determining the correct migration path for your environment, we have created two migration scenarios. The following scenarios are high level examples of the migration process and should be tailored according to the particulars of your Everyplace Suite environment.

The first scenario is a Standard Migration using the currently installed hardware. The second scenario is a High-Availability Migration scenario where high system availability is required. Before beginning migration, be sure to back up your server's operating systems, applications, and databases.

Migration scenarios

- Standard migration
- High-availability migration

Standard migration

The standard migration is adequate if high system availability is not required. Other constraints might also play in the decision to perform a standard migration, such as the lack of available resources to deploy a separate physical environment or the lack of space for a duplicate set of machines.

Perform the following steps for a standard migration:

1. Start Setup Manager. See Prepare for Installation for details on this step.
2. Setup Manager detects existing components and automatically migrates the applicable components to Everyplace Server version 2.1.

High-availability migration

This migration scenario is much more complex than the standard migration. The following assumptions exist for this migration scenario:

- Existing users and administrators can perform most Everyplace Server functions throughout migration,except during user database migration. There may be two small windows of time, perhaps several hours collectively, during which user care and enrollment applications are not available.
- Multiple networks must be used to support the high-availability scenario. The existing Everyplace Suite network is referred to as the production environment.

Everyplace Server version 2.1 is recreated/installed in a physically separate network or cell during the migration period. This alternate network, called the preproduction environment, minimizes downtime and allows all domain names to be left unchanged.

- The high availability scenario requires that most, if not all, components are clustered.

The high availability migration consists of three key phases.

Phase I

1. Establish the number of machines required to support the expected level of traffic during the migration period.
2. Setup the required number of machines and validate connectivity. These machines are the preproduction environment and are used only during the migration period. When migration is complete, the preproduction environment can be freed for other purposes.
3. Migrate databases.
    . Shutdown self-care, customer-care, and enrollment applications in the production environment to prevent uncontrolled changes in the database during the database migration window.
    b. Replicate the SecureWay Directory servers and Tivoli Personalized Services Manager databases that exist on the production environment onto the preproduction environment. For additional information on replicating SecureWay Directory, see Administration Helps in the SecureWay Directory documentation.
    c. Re-enable self-care, customer-care, and enrollment applications in the production domain.
4. Migrate SecureWay Directory (LDAP).
    . Replicate the SecureWay Directory server that exists on the production environment onto the preproduction environment.
    b. Use Setup Manager to install Secureway Directory on the preproduction environment.
    c. Validate the SecureWay Directory install with Directory Management Tool or Suite Manager.
    d. From within the preproduction environment, run the User Migration utility, selecting the option to migrate users.
    e. Use the Directory Management Tool to validate the user migration into the preproduction environment.
    f. Migrate each database that exists on the production domain onto the preproduction environment.
        - DB2 database:
          Use Setup Manager to migrate DB2 in the preproduction environment
          Use DB2 administration console to validate the process.

- Oracle database:
  Oracle requires manual migration.

5. Migrate Tivoli Personalized Services Manager applications.
   . Replicate the Tivoli Personalized Services Manager applications that exist on the production environment onto the preproduction environment.
   b. Run Setup Manager to migrate Tivoli Personalized Services Manager DB2 database schema onto the preproduction environment.
   c. Any non-care applications require manual migration.
   d. To migrate care applications:
      1. Replicate all machines that are running care applications on the existing production environment onto the preproduction environment.
      2. Manually forward-fit any V1 customizations in the production environment to the V2.1 JSPs in the preproduction environment.
6. Use Setup Manager to install the Active Session Table Server in the preproduction environment.
7. For additional components that exist in the production environment:
   . Replicate the component instance in the preproduction environment.
   b. Run Setup Manager in the preproduction environment to migrate the instance.

Phase II

1. Snapshot all SecureWay Directory servers and Tivoli Personalized Services Manager databases that exist on the production environment.
2. Switch users in the production environment to the preproduction environment. The preproduction environment fulfills all functions of the production environment until migration on the production environment is complete.
3. Quiesce the production environment. This command moves the production server to a stopped status.
4. Perform standard migration on the production environment.

Phase III

1. Migrate databases from the preproduction environment to the production environment.
   . Shutdown self-care, customer-care, and enrollment applications in the preproduction environment to prohibit uncontrolled changes in the database during the database migration window.
   b. Snapshot the preproduction SecureWay Directory servers and Tivoli Personalized Services Manager databases to capture changes that may have occurred since launching the preproduction domain.
   c. Use database differentiation tools to integrate any changes from the preproduction environment that's currently running into the SecureWay

Directory servers and Tivoli Personalized Services Manager databases that exist on the production environment.

2. Switch users from the active preproduction environment back to the production environment. This step completes the migration process of the production environment.
3. You may now free up the preproduction environment.

WebSphere Everyplace Server authentication and security

Security can be implemented in a couple of ways in a WebSphere Everyplace Server environment. Authentication can be set up using WebSEAL-Lite or authentication and authorization can be set up using WebSEAL-Lite with Tivoli SecureWay Policy Director. This section provides information on configuring these options.

- Configure WebSEAL-Lite
- Administer WebSEAL-Lite
- Configure Policy Director
- Related information

Configure WebSEAL-Lite

WebSEAL-Lite provides a central point of authentication for the Everyplace Server environment. It also serves as the next non-firewall hop for Wireless Gateway connections and the point of entry for all other connections.

- Edit WebSEAL-Lite configuration settings in SecureWay Directory
- Configure a default realm
- Configure device and network types
- Configure the authentication proxy mode
- Configure WebSEAL-Lite for Reverse Proxy
- Enabling Applications to work with Reverse Proxy
- Considerations when using transparent proxy mode
- Other WebSEAL-Lite Configuration Considerations
- Enabling WebSEAL-Lite for LTPA and Single-Signon

Edit WebSEAL-Lite configuration settings in SecureWay Directory

Common and unique WebSEAL-Lite settings are initially configured during installation. Common entries can be accessed and administered from SecureWay Directory for all instances of WebSEAL-Lite within an Everyplace Server domain after configuring one WebSEAL-Lite within the domain. Unique entries must be configured for each WebSEAL-Lite installed within the domain.

After installation, if WebSEAL-Lite requires a particular configuration setting, it first searches the unique and then the common settings of SecureWay Directory. If the setting is not located within SecureWay Directory, WebSEAL-Lite searches its configuration file ibmwesas.conf.

If the configuration setting is not located in the configuration file, WebSEAL-Lite uses a hard-coded default value for some numeric configuration settings.

To change WebSEAL-Lite settings after installation is complete, you must edit the settings in SecureWay Directory. You can use the Directory Management Tool to locate WebSEAL-Lite settings in SecureWay Directory.

To configure the Directory Management Tool, edit the /etc/dmt.conf DMT configuration file. For example, if the property file contains these lines:

```
#browser=
#tolbar=both
server1.url=ldap://localhost:389
#server1.security.bindDN=
#server1.security.password=
#server1.security.ssl.keyclass=
#server1.secuirty.ssl.keyclass.password=
#server1.admin.url=http://webserver:80
```

You need to set the bindDN and password entries and remove "#" at the beginning of those lines to effect your changes. For example:

```
server1.security.bindDN=cn=adminusr
server1.security.password=pssword
```

The **cid** object indicates the type of WebSEAL-Lite setting. The **cid** for all common WebSEAL-Lite settings is *common*. The **cid** for unique WebSEAL-Lite settings is a generated token with an *eServicePtr* attribute that matches the hostname of WebSEAL-Lite.

The **sys** object indicates the name of WebSEAL-Lite subsystem. The **sys** for all WebSEAL-Lite settings is *wep*.

In order to change a WebSEAL-Lite setting, perform the following:

1. Use Directory Management Tool to open a tree view of SecureWay Directory. Start DMT. Type `dmt` at a command line.
2. Click the **Search** button in the right-hand pane of the tree view to bring up a query screen.
3. Specify the following information to locate the setting:
   - object class: The object class for all WebSEAL-Lite settings is *eProperty*.
   - settingID: Name of WebSEAL-Lite setting you want to change.
4. If the search results in multiple WebSEAL-Lite settings, use the **cid** and **sys** objects to locate the desired setting. For a unique WebSEAL-Lite setting, perform a search on the *eServicePtr* attribute of the **cid** object, which matches the hostname of WebSEAL-Lite.
5. After you locate a WebSEAL-Lite setting, a configuration dialog is displayed.
6. Edit the **cisProperty** field to configure WebSEAL-Lite setting.

   Note: You must use the correct format to configure WebSEAL-Lite settings. The **cisPropertyType** field indicates the format you use to configure the setting.

   A description of WebSEAL-Lite settings, the settingID, and the cisPropertyType for each setting is provided in Everyplace Server authentication properties for WebSEAL-Lite.

Configure a default realm

WebSEAL-Lite permits a default realm to be configured in the ibmweas.conf file. The presence of a default realm allows users to enter their user names at the Everyplace Server login prompt rather than entering an explicit realm. The presence of a default realm alleviates users logging in to Everyplace Server from having to enter an explicit realm in the their user names at login prompts. To create the `default realm`, enter the `default_realm` directive in the ibmwesas.conf file, for example, `default_realm acme.com`

Configure device and network types

WebSEAL-Lite identifies devices associated with incoming requests based on mapping rules defined in the device profiles stored in SecureWay Directory (see the *WebSphere Transcoding Publisher Administrator's Guide* for information about configuring device profiles and the device mapping rules). If WebSEAL-Lite cannot determine a device type based on the available mapping rules, WebSEAL-Lite uses a default network or device type to implement transcoding. The default network type is wireless. The default device type is default.

To change the default network or device type, open the file where the SecureWay Directory parameters are stored and search for the following:

```
default_device_type
```

or

```
default_network_type
```

You can select one of the following options for the default device type:

- WML-Device: Any WAP-compliant device (pre-defined default)
- Palm-Pilot3 HandWeb11: Palm Pilot HandWeb Browser
- NT.InternetExplorer4: MS Internet Explorer browser (version 4)
- NT.InternetExplorer: MS Internet Explorer browser (version 5)
- NT.Netscape45: Netscape Navigator browser (version 4+)
- WinCE.PocketIE20: MS Windows CE-compatible device
- I-Mode 501: Wireless Phone - I-Mode Model 501i
- I-Mode 2 Color Phone: Wireless Phone - I-Mode 2 Color Model
- I-Mode 2 Monochrome: Wireless Phone - I-Mode 2 Monochrome Model

Notes:

- The device type names are case-sensitive.
- Any valid device type as defined in the WebSphere Transcoding Publisher device preference profiles is allowed.

You can select one of the following options for the default network type:

- wireless: Any wireless network
- dial: Dial-up network
- default: Direct-connected or LAN

Note: Generally, you should choose a network type with the lowest bandwidth that your enterprise supports as the default.

Configure the authentication proxy mode

WebSEAL-Lite may be configured in one of three modes or roles using the 'AuthServerRole' directive in the ibmwesas.conf file:

Authentication Proxy mode
In this mode, WebSEAL-Lite functions as a reverse proxy and issues HTTP 401 challenges whenever Basic Authentication is used. In this configuration, the authentication proxy accepts client requests, then routes those requests to another server. The authentication proxy appears to the client to be the content server, and the client is not aware that the request has been sent to another server. In addition, all SSL (Secure Sockets Layer) connections are terminated at the reverse proxy. You must configure the authentication proxy as a reverse proxy within the ibmproxy.conf file by making use of mapping directives, or by using the WebSEAL-Lite junctioning capability. For more information regarding reverse proxy configuration, refer to the

WebSphere Edge Server Caching Proxy (Web Traffic Express) User's Guide. For more information regarding junctions refer to the WebSEAL-Lite User's Guide.

Transparent Proxy Mode
In this mode, WebSEAL-Lite functions as a forward proxy and issues HTTP 407 challenges whenever Basic Authentication is used. In this configuration, the authentication proxy accepts client requests and tunnels them through to the intended back-end server. The client must configure an HTTP proxy. In addition, SSL requests are tunneled through to the back-end server. No special proxy mapping or junctioning is required in this mode.

Detecting Mode
In this mode, the Authentication Proxy dynamically determines, based on the type of request from the browser, whether to treat the request as a reverse proxy or forward proxy request. Using this mode, only a single instance of Authentication Proxy need be configured to handle both requests for internet content as well as reverse proxy content.

There are two sample configuration files available on Disc 10 to help with configuring WebSEAL-Lite for both Authentication proxy and Transparent proxy modes. There is one each for both the authentication proxy mode and for the transparent proxy mode. The files are:

Authentication proxy sample file: `ibmproxy_ap.conf`
Transparent proxy sample file: `ibmproxy_tp.conf`

The file entries that pertain to Everyplace Server are designated by "WES" in the associated comments.

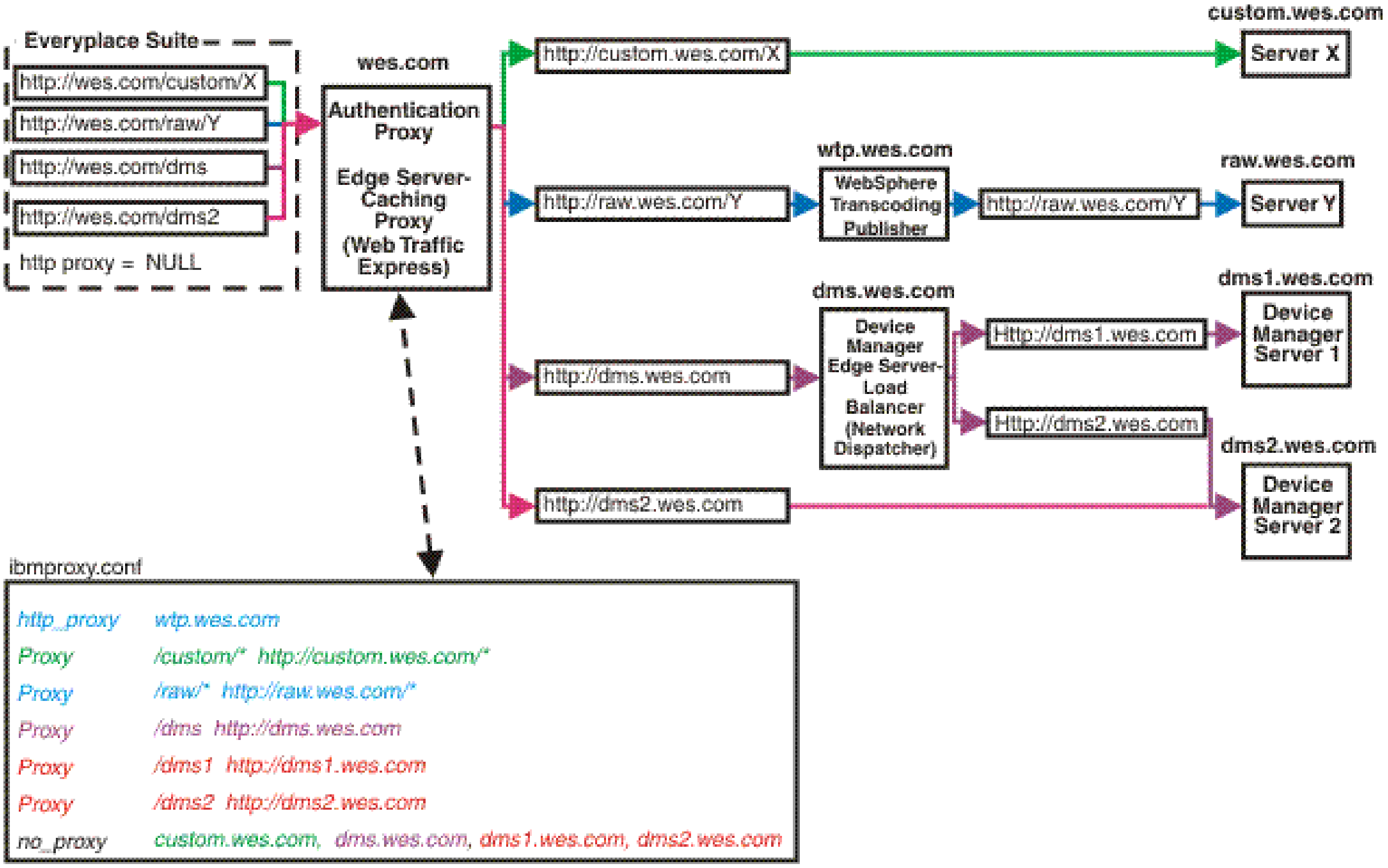
Configure WebSEAL-Lite for Reverse Proxy

Configuration for reverse proxy involves providing the mappings from virtual URIs to "real" back-end URIs and servers as well as specifying the type of secure connections required for those back-end connections. Such configuration also requires specifying which destinations require intermediate routing through a proxy.

There are two basic ways to configure WebSEAL-Lite as a reverse proxy:

1. Utilize the WebSEAL-Lite URL mapping, routing, and junction capability. This capability is documented in the WebSEAL-Lite User's Guide and the sample osdef.conf configuration file. Using WebSEAL-Lite mapping and junctions often greatly reduces the amount of mapping required. Most times, only a single mapping definition is needed for a given back-end Web site. Intermediate proxy routing can also be specified on a junction-basis.
2. Utilize the Caching Proxies "Proxy" mapping directives in the ibmproxy.conf file. Since explicit mappings are required for all URLs that can be accessed on a given Web site, more configuration is required than for WebSEAL-Lite routing. Such control might be desirable. Only a single intermediate proxy can be specified for all back-end servers using the ibmproxy directives, so if more granular routing is needed, use the WebSEAL-Lite proxy routing. The following diagram illustrates a simple configuration and the ibmproxy.conf directives required to achieve it.

Configure the authentication proxy as a reverse proxy in ibmproxy.conf



```
http_proxy   wtp.wes.com
Proxy        /custom/*  http://custom.wes.com/*
Proxy        /raw/*  http://raw.wes.com/*
Proxy        /dms  http://dms.wes.com
Proxy        /dms1  http://dms1.wes.com
Proxy        /dms2  http://dms2.wes.com
no_proxy     custom.wes.com,  dms.wes.com,  dms1.wes.com,  dms2.wes.com
```

Enabling Applications to work with Reverse Proxy

Setting up general applications to work with Reverse Proxy

WebSEAL-Lite performs authentication and reverse proxy routing for all requests through the Everyplace Server domain. In order to enforce strong authentication and security while enabling single-signon for applications that reside behind WebSEAL-Lite, all traffic through the Everyplace Server domain must be directed through WebSEAL-Lite via normal internet routing. Such applications must follow certain rules in order to enable the correct routing. In particular, applications using any of the following HTTP/HTML mechanisms:

- BASE tag - specifies the BASE URI to use for referenced documents
- Location: headers in Redirection responses - specifies the domain to which HTTP redirection must occur
- Absolute URI references (images, hrefs, etc.) - any full-path reference which includes the server name

must specify in these references the name of the WebSEAL-Lite proxy server or the WebSEAL-Lite server cluster name (if Network Dispatcher is used).

Setting up the TISM user applications to work with Reverse Proxy

Tivoli Internet Services Manager applications (TISM) (enrollment, self-care, and customer-care) utilize the BASE tag in their JSP definitions and must be set up as described above. See the parameter value for BASE URI reference in the HSMCommon.properties file (on AIX in `/usr/TivTSM/classes` and on Solaris in `/opt/TivTSM/classes`) that allows you to specify the domain name of the WebSEAL-Lite server that is used in all BASE tag references.

You must also edit the following files:

1. Modify HSMCommon.properties located in /usr/TivTSM/classes on AIX or /opt/TivTSM/classes on Solaris with the appropriate information
   - Reverse Proxy Settings

- If a Reverse Proxy is in use, set `ReverseProxyActive=true`
- The servlets will then use the specified `Scheme`, `ServerName`, and `ServerPort` to set the URL base rather than using information received from the HTTP header. For example,

  ```
  ReverseProxyServerName=Proxy.Name
  ReverseProxyServerPort=443
  ```

2. When using WebSphere Application Server, modify DefaultAuthenticator.properties located in /usr/TivTSM/authentication/servlets, /usr/TivTSM/HSMSelfcare/servlets, and /usr/TivTSM/perso/servlets on AIX and /opt/TivTSM/authentication/servlets, /opt/TivTSM/HSMSelfcare/servlets, and /opt/TivTSM/perso/servlets on Solaris. Set the following properties as follows:
   - `authentication.multidomain.enabled = no`
   - `authentication.type1 = username`
   - `authentication.type1.header = X_IBM_PVC_User`
   - `authentication.type2 = basic`
3. Modify enroll.html, selfcare.html, and custcare.html to point to the Reverse Proxy server exactly as it was specified in the HSMCommon.properties file. Include the proper scheme, name, and port. These files are located in http_root/htdocs/tsm where http_root is the root directory for IBM HTTP Server.

Setting up DMS to work with Reverse Proxy

DMS utilizes redirection during device management operations when there are multiple device management servers (managed by Load Balancer) which can service the request. The redirected requests must be routed back through WebSEAL-Lite. In order to enable this capability, you must specify a parameter value for an authentication proxy URL, authProxyDmsUrl, during creation of the Device Manager server, which occurs during the configuration of the WebSphere Application Server. For additional details on how Device Manager handles authentication and redirection, refer to *Planning for Tivoli Personalized Services Manager*.

Considerations when using transparent proxy mode

If you install WebSEAL-Lite in transparent proxy mode, you must set the server address and port number of the Edge Server Caching Proxy in the Netscape Navigator HTTP proxy setting panel before attempting to open the Caching Proxy from Everyplace Suite Manager.

Other WebSEAL-Lite Configuration Considerations

If you install WebSEAL-Lite, the Everyplace Server Setup Manager prompts you for configuration information. Some of this information is common across all instances of WebSEAL-Lite within the Everyplace Server domain. Other information is unique to each instance of WebSEAL-Lite. After installation, you can reassign these configuration parameters from common to unique or from unique to common. You must provide the following common information:

- Primary Active Session Table server name
- Secondary Active Session Table server name
- Maximum session age (minutes)
- Default retry after delay (seconds)

You must provide the following information for each server installing WebSEAL-Lite. This information is unique to each server.

- AuthServerRole

- Maximum session cache size
- Active Session Table daemon cleanup interval

Note: When configuring WebSEAL-Lite to deploy WebSphere Transcoding Publisher in your Everyplace Server domain, there may be performance penalties when routing content through Transcoding Publisher. Be careful to only use transcoding when it is needed.

Enabling WebSEAL-Lite for LTPA and Single-Signon

You need to set up a key file for WebSEAL-Lite when configuring it for Lightweight Third Party Authentication (LTPA) support. You can use the same key file created for WebSEAL-Lite in WebSphere Application Server, but you need to edit the key file to replace `ltpa.*` with `WASPrefix.*`. Where `WASPrefix` is the prefix used by WebSphere Application Server.

Alternatively, you can generate the key file using WebSphere Application Server, supplying the key file and password to WebSEAL-Lite. After the key file has been generated by WebSphere Application Server, it should be edited from `WASPrefix`.* to `ltpa.*`

To create the key file for WebSEAL-Lite, issue the following commands:

```
export CLASSPATH=/opt/pdweb-lite/classes/wesltpa.jar
java com.ibm.wte.sso.TestSSO <KeyFilename> <password>
```

Administer WebSEAL-Lite

- Debug WebSEAL-Lite
- WebSEAL-Lite operations tasks
- Everyplace Server authentication properties for WebSEAL-Lite

Debug WebSEAL-Lite

WebSEAL-Lite contains two modules: `wesauth.so` and `wesauth.so.debug`. Use the `wesauth.so.debug` module to debug WebSEAL-Lite. In order to switch to the debug module, rename the `wesauth.so` module to any desired name, and then rename the `wesauth.so.debug` module as `wesauth.so`.

The `ibmwesas.conf` file should have the following statement to enable debugging: `debugLevel 10`. This statement should be the first statement in the file to allow debugging for processing the rest of the file. The number 10 specifies a level which tells how much output will be given. A higher number specifies more output.

WebSEAL-Lite Operations tasks

WebSEAL-Lite authserv command line utility can be used to control and monitor WebSEAL-Lite plug-in while it is running. For security reasons, the utility must run on the same system as WebSEAL-Lite. The *authserv* utility lets you perform certain operations tasks on WebSEAL-Lite, including suspending and resuming WebSEAL-Lite, refreshing configuration settings, and displaying operational statistics.

The *authserv* program has the following syntax:

```
authsrv [-?| -help] [-p:serverPort] [funcname[args,,,]] ]
```

Arguments:

- --? or --help: Displays help information for the program. Issuing the *help* function name displays a list of functions supported by the server.

- --p: Sets the TCP port on which to connect to the server. The port is configured in *ibmwesas.conf*. A default port of 9734 is used.
- *funcname*: The function to be executed on the server. See the table below for more information.
- *args*: Any arguments for the function.

If you specify a function name on the command line, that function is sent to the server. Once the response and output is received, the client terminates the session and exit.

Note: Not all configuration files are reread when WebSEAL-Lite is refreshed.

If you do not specify a function name or help options, the program establishes a session with the server, enter a command loop reading from standard input (stdin), and wait for you to enter a function name with optional arguments. You can exit the program by specifying one of the following function names:

```
quit
```

or

```
exit
```

The operations tasks that you can perform using the authserv program are shown below:

Operations tasks for the authserv program

Function Name	Argument(s)	Description
suspend	[Retry-After delay (seconds)]	Stops normal operation of WebSEAL-Lite instance. Any requests received while suspended are rejected with a 503 status code specifying a Retry-After value as passed in. If no value is specified, then the default Retry-After value is used.
resume		Begins normal operation of WebSEAL-Lite instance after suspension.
refresh	[configuration file path]	Updates the WebSEAL-Lite active configuration settings from ibmwesas.conf or SecureWay Directory by forcing a read of ibmwesas.conf or SecureWay Directory. If noibmwesas.conf configuration file is specified, only a SecureWay Directory read is performed. Refresh performs an implicit suspension of WebSEAL-Lite. The Caching Proxy must be shut down separately to refresh the Caching Proxy settings or other WebSEAL-Lite settings such as osdef.conf file settings.
display		Returns operational statistics.
flush		Flushes the session cache for all users.
start_daemon		Starts the active session table clean-up daemon.
stop_daemon		Stops the active session table cleanup daemon.
help		Returns a list of functions supported by the server.
quit		Causes the session with the client to be terminated.

If an operations command fails, information on the failure is reported to either the command line console or to the Caching Proxy console.

Everyplace Server authentication properties for WebSEAL-Lite

A description of WebSEAL-Lite settings, the settingID, and the cisPropertyType for each setting is provided here. These properties can be configured using SecureWay Directory or in the WebSEAL-Lite `ibmwesas.conf` file.

LDAP SettingID / Parameter	LDAP cisProperty Type / Value type	Description
MaxSessionAge	Number	The maximum amount of time, in minutes, a WebSEAL-Lite maintains information about a user's session. After this time has passed, WebSEAL-Lite clears the information. WebSEAL-Lite resources, such as disk space, are used while the information is maintained. But once the information is cleared, additional time may be necessary to recreate it.
DefaultRetryDelay	Number	Default Retry After Delay is the default amount of time, in seconds, the device should wait to retry the request.
AuthServerRole **Note:** This parameter can only be read from the WebSEAL-Lite `ibmwesas.conf` file.	Literal	WebSEAL-Lite may serve as one of three proxy types or roles: as an authentication proxy that intercepts all requests made to resources within the Everyplace Server domain, or as a transparent authentication proxy that allows access to content provided by third-party content servers while taking advantage of Everyplace Server authentication and transcoding. AuthServerRole can also automatically detect the role of authentication or transparent proxies.
ASTServerP	Literal	The host name of the primary server that is used to manage the active session table and its entries.
ASTServerS	Literal	The host name of the backup server, if applicable, that is used to manage the active session table and its entries.
MaxSessionCache	Number	The maximum size of the local in-memory cache for a particular proxy.
ASTCleanupInterval	Number	The amount of time, in seconds, in which session information should be cleared from the active session table server.

Configure Policy Director

Policy Director implements finer levels of access control rights to web applications. This can be done by integrating Policy Director with Everyplace Server components. Policy Director verifies a user's authorization to enter a restricted area and delivers or restricts page access based on the user's access level.

- Configure Policy Director for authentication

- Customize the pd_admin.ksh file for Everyplace Server

Configure Policy Director for authentication

If Policy Director is used for authentication, WebSEAL-Lite does not check the Everyplace Server user status bit (ibm-tismStatus) when the user's status changes. Policy Director provides its own analogous bits for this function: `secAcctValid`, which indicates if the user's account is valid, and `secPswdValid`, which indicates if the user's password is valid.

Because Everyplace Server does not provision these fields when the user's status changes, you must update the Policy Director account status field whenever Customer Care is customized. Any changes to a user's status driven by the SUBSCRIBER_STATUS_CHANGE or CHILD_SUBSCRIBER_STATUS_CHANGE events should be provisioned to Policy Director's account status for the user. If the Policy Director account status field is not updated, the disconnected Tivoli Internet Services Manager user could be authenticated and authorized since the Tivoli Internet Services Manager status is not provisioned to Policy Director.

Customize the pd_admin.ksh file for Everyplace Server

Tivoli Internet Services Manager (TISM) provides a "bridge" process that automatically creates, updates, and deletes user information in Policy Director based on corresponding changes in TISM (Enroll, place a Customer Order, change Deal, is deleted, etc.) and also associates the user with the correct Authorization groups in Policy Director. Edit the `pd_admin.ksh` script file (located in `/usr/TivTSM/provisioning/clients/pd/bin` on AIX and `/opt/TivTSM/provisioning/clients/pd/bin` on Solaris) to configure and customize the bridge. You need to make the following updates to get the bridge to work in an Everyplace Server environment.:

1. Edit the LDAP suffix to match the LDAP suffix in your environment. For example,
   `LDAP_SUFFIX="o=lagaude,c=us"`
2. Edit the User creation command to match the Everyplace Server LDAP DIT convention. It must be specified as:
   ```
   USER_CREATE_CMD="user create -no-password-policy $tpd_username
   uid=$username,cn=users,ou=$realm,$LDAP_SUFFIX"
   ```
3. You may also need to update the following values:
   - Full path name of the PolicyDirector admin command.
     `PDADMIN=/opt/PolicyDirector/bin/pdadmin`
   - PolicyDirector admin login/password. `LOGIN="-a sec_master -p more2know"`
   - Full path name of the log file. `LOG_FILE="/var/adm/logs/pd_admin.log"`

Related information

- WebSEAL-Lite
- Tivoli SecureWay Policy Director

Configure Everyplace Wireless Gateway

The following configuration tasks can be performed on Everyplace Wireless Gateway:

- Enable Everyplace Wireless Gateway to access RADIUS authentication
- Configure Wireless Gateway during creation of the gateway resource
- Configure Wireless Gateway after creation of the gateway resource
- Set user classification flags

Enable Everyplace Wireless Gateway to access RADIUS authentication

By default, installation of Everyplace Wireless Gateway does not include RADIUS authentication. If the added security of RADIUS authentication is desired, you can enable Wireless Gateway to access RADIUS authentication. This configuration adds Everyplace Wireless Gateway to the RADIUS authentication as a client and enables Everyplace Wireless Gateway to access the RADIUS server.

To add Wireless Gateway as a RADIUS client, perform the following:

1. The NAS Password must be used as SHAREDSECRET in the Wireless Gateway configuration.
2. Use your favorite LDAP browser utility to located the `radiussharedsecret` attribute. Below is the directory structure.
   ```
   radiussharedsecret: <entry>
   dc=com
   dc=<your suffix>
   sys=SDP
   sys=ewg
   cn=root
   ou=<default resources>
   cn=<hostname>
   ```
3. Set NAS Password to reflect `radiussharedsecret` entry.

Configure Wireless Gateway during creation of the Gateway resource

During installation of Wireless Gateway, Setup Manager prompts you for the required WAP and RADIUS information, such as primary and secondary IP addresses and SHAREDSECRET. You may perform the following configuration steps when creating a Gateway resource for Everyplace Wireless Gateway in an Everyplace Server environment:

1. Configure Wireless Gateway to authenticate using RADIUS, if additional security is desired. The default selection is off.
2. Configure both Wireless Access Protocol (WAP) and non-WAP resources to prevent user validation. The default for WAP and non-WAP resources is to request user validation.
3. For the WAP gateway resource, set the host running WebSEAL-Lite as the HTTP proxy for WAP gateway resource.

Note: The HTTP proxy defined in the user records will supercede the gateway property setting.

Refer to the *Wireless Gateway Administrator's Guide* for information on creating a Gateway resource.

Configure Wireless Gateway after creation of the Gateway resource

You may perform the following configuration steps after installing Everyplace Wireless Gateway and creating the Gateway resource:

1. Use the Tivoli Internet Services Manager Console to add a RADIUS client that enables Everyplace Wireless Gateway to access the RADIUS server. To do this:

    . Start the Internet Services Manager Console.

    b. Select **Define Network Access**, then expand the Define Network Access tree view.

    c. Select **Clients**.

    d. Right-click **Clients**, then select **Create**.

    e. Enter the following data:

    - NAS IP and NAS ID=Everyplace Wireless Gateway IP Address
    - NAS Name=Everyplace Wireless Gateway Host Name
    - NAS Password=Value of `radiussharedsecret`
    - Select the default values for the remaining arguments.

f. Select **OK**.

g. Restart the RADIUS Server as follows:

i. From the Tivoli Personalized Services Manager RADIUS server, enter the command:
`cd /usr/TivTSM/radius/bin`

ii. Issue the `./reload_radius_db2.ksh` script (for Oracle database, use the command: `./reload_radius.ksh`).

2. Specify a SHAREDSECRET setting for the RADIUS server. See Enable Everyplace Wireless Gateway to access RADIUS authentication for information on locating the radiussharedsecret entry.

Note: If a user PPP authenticates through Wireless Gateway, the user is authenticated, and the session remains usable for WAP services withough an additional challenge to the WAP user-agent. Wireless Gateway writes an Active Session Table record for the login that WebSEAL-Lite uses to authenticated WAP traffic from the originating IP address.

Set user classification flags for Everyplace Wireless Gateway

Two fields are included in the WebSphere Everyplace Server LDAP schema which can be used to classify users of the suite. These fields can be set either TRUE or FALSE. This classification might allow fewer users to be displayed when using Everyplace Wireless Gateway's Gatekeeper tool to display users since the Gatekeeper tool uses these two fields to reduce the scope of user list queries. The fields are:

- **ibm-WGclient**: this field classifies a user as a user of Everyplace Wireless Gateway. Gatekeeper displays only users with this field set to TRUE.
- **ibm-WAPclient**: this field classifies a user as a wireless access protocol (WAP) user. Gatekeeper displays WAP characteristics only for users with this field set to TRUE.

By default when a user is enrolled, these two fields are set to TRUE by the Tivoli Internet Services Manager LDAP provisioning gateway (LDAPGateway) via a setting in the `LDAPGateway.properties` file.

You can customize the setting of these fields using techniques such as defining user profile extensions and using the LDAPGateway to provision the user profile extensions. See the Tivoli Internet Services Manager documentation for the LDAPGateway and User Profile Extensions for more details.

Third-party gateway support

Everyplace Server allows the use of third-party WAP gateways, which are WAP gateways other than Everyplace Wireless Gateway, if desired. The use of third-party gateways is supported by WebSEAL-Lite.

WebSEAL-Lite searches SecureWay Directory for third-party gateway definitions and uses that information to invoke a customer-supplied, gateway-specific C or C++ plugin routine that returns a client ID (for example, a phone number or device number) that identifies the user who initiated the HTTP request coming through the gateway. WebSEAL-Lite uses the client ID to search the *cn=users* portion of SecureWay Directory for the ePerson entry whose client ID attribute matches the client ID returned by the plugin. The client ID attribute is configurable and is contained in the third-party gateway definition.

The plugin has access to Edge Server Caching Proxy (Web Traffic Express) APIs that return a specific HTTP header, the URL, the query string, and other information about the current request. If a matching ePerson entry is found, its user ID and organizational unit attributes are used to construct the ID placed on the X-IBM-PVC-User HTTP header, and the client ID is placed on an X-IBM-PVC-Client-id HTTP header. Both of these headers are then available for use by downstream components.

WebSEAL-Lite invokes the appropriate plugin when handling a request from a third-party gateway. The actual gateway-specific plugins is written by IBM, the gateway providers, or customers.

This section covers the following topics regarding third-party gateway support:

- Installing third-party gateway support
- Unique client ID support
- Performance considerations
- Customer-supplied, gateway-specific client ID plugin

Installing third-party gateway support

Install third-party gateway support as follows:

1. Code the plugin routine for each third-party gateway you plan to use. Create a shared library containing the plugin and place it in an appropriate directory with appropriate file ownership and permissions. Shared libraries typically have an `.so` file extension and reside in `/usr/lib/`. Detailed plugin information is shown in Customer-supplied, gateway-specific client ID plugin.
2. Shut down Caching Proxy so that WebSEAL-Lite is no longer running. Make a backup copy of your existing WebSEAL-Lite shared library. The library is shipped with Everyplace Server as `wesauth.so` and is typically installed in `/usr/bin/`. Replace the library with `wesauth.so.debug` if you want to capture additional debugging information in the Caching Proxy log files. The file ownership and permissions must match those of the original `wesauth.so` library.
3. Choose a location in SecureWay Directory for each type of third-party gateway you plan to use.

   Note:

   - You do not have to choose a SecureWay Directory location for each instance of a particular type of third-party gateway.

   This location must be under the baseDN (the Distinguished Name) specified in your WebSEAL-Lite configuration file, but **NOT** under the *cn=root*, *sys=ewg* entry, because that section is reserved for use by Everyplace Wireless Gateway.

   The configuration file is usually called `ibmwesas.conf`, and is installed in the following locations:

   - For AIX:

     `/usr/lpp/IBMEPS.Auth/`

- For Solaris:

  ```
  /opt/IBMEPSAu/
  ```

For each gateway, add a gatewayDN statement to your WebSEAL-Lite configuration file. This statement specifies the gateway's location in the SecureWay Directory tree, relative to the baseDN. The syntax for this statement is:

```
gatewayDN DN_relative_to_baseDN
```

(for example: `gatewayDN cn=mygateway`).

4. Add the *wlGateway* and *wlMni* object classes if they are not already located within your SecureWay Directory schema. Use the SecureWay Directory administration Web pages to make a backup copy of the SecureWay Directory tree and schema (LDIF). Run the `wg_schema` program to add these object classes. To access the SecureWay Directory administration Web pages, point your browser to `/ldap` on the machine running the SecureWay Directory server (for example `http://dirserver2.company.com/ldap`). You must have a Web server running on the same machine as the SecureWay Directory server. The syntax is:

   ```
   wg_schema -h <ldap_server_hostname> -D <admin_DN> -w <admin_pw>
   ```

5. Use the Directory Management Tool (see Web site *http://www-4.ibm.com/software/network/directory/library/publications/31/dmt/dparent.htm* for instructions on using the Directory Management Tool). Add entries to the SecureWay Directory as follows:

   . Add a container entry at the spot in the tree you identified on the *gatewayDN* statement in the configuration file. The RDN (Relative Distinguished Name) for this entry must match that on the *gatewayDN* statement. Continuing with the example above, you would use *cn=mygateway* as the RDN.

   b. Under the container entry add an ePropertySet. A typical RDN for this entry might be *cid=Common*, but the RDN is not important.

   c. Under the ePropertySet add the following eProperty entries:

      An entry with an RDN of *settingID=pluginName*. Set the entry's *cesProperty* attribute to the name of your plugin for this gateway. The plugin name must be of the form: `library_name:function_name` where *library_name* is the name of the shared library that contains `function_name`. An example plugin name is `myLibrary.so:getClientID`.

      An entry with an RDN of *settingID=clientIDAttributeName*. Set the entry's *cisProperty* attribute to the name of the attribute in the ePerson object that you use to locate, in SecureWay Directory, the user associated with the client ID returned from your plugin. For example, if your plugin returns an MSISDN number, you could set this eProperty to internationalISDNNumber.

      (optional) An entry with an RDN of *settingID=pluginToken*. Set the entry's *cesProperty* to a string token that you want passed to your plugin. Any use of the string token is defined solely by your plugin. For example, it could be used to pass in the name of an HTTP header, a gateway type, or a plugin version number.

   d. Under the **container** entry, add *wlGateway* or *wlMni* entries to identify specific instances of the type of third-party gateway that you have installed. The RDN is not important (for example, `cn=mygateway1`). When adding a *wlGateway*, specify a structural object class of *wlGateway* and an auxiliary object class of *ibm-wlResource*. When adding a *wlMni*, specify a structural object class of *wlMni* and an auxiliary object class of *ibm-wlResource*.

      The *wlGateway* entry identifies a single gateway machine, while the *wlMni* entry can identify a range of IP addresses running such gateways. If you add a *wlGateway* entry, you must specify its hostname or IP address in the host attribute. If you add a *wlMni* entry, you must specify an IP address in the *netaddr* attribute and a mask in the *netmask* attribute; the mask is applied to the IP address to define a range of IP addresses.

6. Restart Edge Server Caching Proxy. If you installed the debug version of `wesauth.so`, you can set the debug message level via the `debugLevel` statement in the WebSEAL-Lite configuration file to provide debugging/progress information in one of the Caching Proxy log files.
7. As with all SecureWay Directory information used by WebSEAL-Lite, if you change any of the SecureWay Directory information used in WebSEAL-Lite's third-party gateway, you must either issue an `authsrv refresh` command or stop and restart Caching Proxy. If you don't, the changes might not have any effect on WebSEAL-Lite operation.
8. You must now integrate ClientID support into the Tivoli Personalized Services Manager applications to assign unique IDs to Everyplace Server users. For example, you could configure Customer Care to use ClientIDs as follows:

   . For AIX, update the `/usr/TivTSM/custcare/content/JPanSearch.cfg` file to modify the following two lines.

   ```
   SP_Defined_1=MSISDN
   SP_Defined_2=
   ```

   b. Save the file.

   c. Re-start the customer care servlet using the WebSphere adminclient tool.

   d. When a customer service representative logs into CustomerCare, they have the option to add or modify the value of the **MSISDN** field of any user.

   Note:

   ■ **MSISDN** is the name of the field where the client ID is specified. You may give this field another name (for example, **ClientID**).

   e. For Solaris, update the `/opt/TivTSM/custcare/content/JPanSearch.cfg` as shown above and follow the procedure used for AIX.

Unique Client ID support

If using client ID as the third-party authentication mechanism in the Everyplace Server environment, you must add further validation to the Tivoli Personalized Services Manager sub-components to ensure that client IDs are unique. By default, there is no data validation of end user input for client ID, which allows duplicate or incorrect client IDs to be created. For additional information about validating profile extensions, please contact your IBM service representative.

WebSEAL-Lite creates an X-IBM-PVC-Client-ID header containing the value returned from the plugin if that header does not already exist in the HTTP stream. Some Everyplace Server functions, such as Location Proxy, depend on this header to work correctly. Therefore, the plugin must return the correct value format. The format is as follows:

```
X-IBM-PVC-Client-Id
```

This provides additional information to uniquely identify the user's session and/or device. The value is a token that represents a single device and depends on the connection type and/or connecting gateway. The token consists of a variable-length string that is the device identifier in the format dictated by the type indicator and a two-digit type indicator which can be:

- `08` - indicates the client ID is an IP address
- `1F` - indicates the client ID is a phone number (caller number or MSISDN)
- For example, `x-ibm-pvc-client-id: 089.37.58.55` or `x-ibm-pvc-clientid: 1F0119195551212`

Performance considerations

To improve the performance of the SecureWay Directory search for ePerson entries with a client ID attribute that matches that returned from your plugin, you can create a DB2 index for that relationship. WebSEAL-Lite caches the result of these lookups in memory so that not every request from your third-party gateways involves an SecureWay Directory search, but the DB2 index is still beneficial to performance.

This support adds one or more caches for SecureWay Directory information mapping client IDs to user information. The size and cleanup interval of these caches is the same as for the active session table cache and is controlled by the *MaxSessionCache* and *ASTCleanupInterval* configuration values, respectively. These new caches are flushed when an `authsrv flush` or `authsrv refresh` command is issued.

Since the plugin is called for every request coming through a third-party gateway, it is important that the plugin be efficient and not perform lengthy operations unless absolutely necessary.

Customer-supplied, gateway-specific client ID plugin

The customer-supplied gateway-specific plugin is responsible for returning a client ID for the current request. This routine runs as part of the same process and threads as WebSEAL-Lite. Because Caching Proxy plugins run in a multithreaded environment, the gateway-specific plugin must be thread-safe. The plugin has access to the HTTP headers, URL and query string, and other information about the current request through the same Caching Proxy plugin APIs as WebSEAL-Lite. API `httpd_getvar()` is used to obtain the request information.

The plugin can use `HTTPD_log_error()` to write information to the Caching Proxy Web Traffic Express error log, which is identified on the ErrorLog directive in the Caching Proxy configuration file. Caching Proxy API information is available in the *Web Traffic Express Programming Guide* available from: http://www.ibm.com/software/webservers/edgeserver/library.html

You can have multiple plugins in the same library and coded in the same source code file. However, they must have different function names.

Interface

The plugin receives a buffer into which it must place the client ID. The plugin must also set a return code indicating its success or failure.

The C++ prototype for this function is:

```
extern "C" int function_name(const unsigned char *logHandle, char *buffer, int
*bufferSize, const char *token);
```

The C prototype for this function is:

```
int function_name(const unsigned char *logHandle, char *buffer,
int *bufferSize, const char *token);
```

The variables you should enter are defined as follows:

- logHandle: handle to the WTE log. Used on WTE APIs only.
- buffer: pointer to a buffer provided by the caller to hold the client ID upon return from this function
- bufferSize: pointer to a variable containing the size, in bytes, of the buffer provided by the caller
- token: pointer to a string token that was fetched from the *pluginToken* eProperty in this gateway's configuration section in SecureWay Directory. This token's content, format and use are under complete control of the plugin. If the *pluginToken* eProperty is not defined in SecureWay Directory or has no value, the token argument is NULL when this plugin is called.

Output is shown in the form of the following return codes indicating the function's success or failure:

- 0: The client ID was obtained and copied to the caller-supplied buffer as a null-terminated string.
- 1: The caller-supplied buffer is too small to hold the entire client ID. When returning this code, this function must first update the size variable pointed to by the *bufferSize* property to contain the size of the buffer required to hold the result. In that case, the caller acquires a buffer of the requested size and calls this function again.
- 2: This function was unable to return the client ID. No specific reason is implied by this return code.
- 100-199: These return codes are reserved for plugin use. They may be used to indicate a specific error instead of returning return code 2. They are treated the same as return code 2 by the caller.

Creating the shared library

The plugin must reside in a shared library accessible to WebSEAL-Lite during request processing. A shared library containing a C++ plugin can be created using compile and link commands similar to the following:

- C++ plugin using IBM CSet++ on AIX:

  ```
  xlC_r -c myPlugin.cxx -I/usr/samples/internet_server/API
  makeC++SharedLib -p0 -o myPluginLibrary.so
  -bI:/usr/samples/internet_server/API/libhttpdapi.exp myPlugin.
  ```

- C++ plugin using Sun Workshop on Solaris:

  ```
  CC -c -mt myPlugin.cxx -I/usr/samples/internet_server/API
  CC -G -mt -o myPluginLibrary.so myPlugin.o
  -L/usr/samples/internet_server/API -lhttpdapi
  ```

- C plugin using IBM CSet++(R) on AIX:

  ```
  xlc_r -c myPlugin.c -qcpluscmt -I/usr/samples/internet_server/API
  makeC++SharedLib -p0 -o myPluginLibrary.so
  -bI:/usr/samples/internet_server/API/libhttpdapi.exp myPlugin.o
  ```

- C plugin using Sun Workshop on Solaris:

  ```
  cc -c -mt myPlugin.c -I/usr/samples/internet_server/API
  cc -G -mt -o myPluginLibrary.so myPlugin.o
  -L/usr/samples/internet_server/API -lhttpdapi
  ```

These commands create a shared library called `myPluginLibrary.so` from a single source file (`myPlugin.cxx`) using Caching Proxy API files in the `/usr/samples/internet_server/API` directory.

Configuring Load Balancer for WAP Devices

Restrictions prevent WAP-enabled devices from receiving content that exceeds the device's storage capacity. The length of many HTML documents typically exceeds the device's capacity. This section includes information on how WebSphere Everyplace Server components interact to deliver extensive data to wireless devices. The following overview explains some of the concepts involved in effectively achieving this goal.

- Fragmentation
- Content Based Routing (CBR)
- Cookie affinity
- Considerations when running Transcoding Publisher behind WebSEAL-Lite
- Related information

Fragmentation

Fragmentation resolves the device storage problem by splitting an oversized file into units smaller than the maximum size limitation. In addition, the fragmentor adds links to each unit generated to allow navigation. The maximum size is a property of the device, so the fragmentor uses the device information retrieved from the device preference profile to determine the maximum size value. If the file exceeds the maximum allowed size, then the fragmentation process begins. The final step in the fragmentation process is link rebinding, during which the link attributes are adjusted to reflect the correct target name.

The fragmentation engine stores non-first fragments in a general purpose "resource repository," a Transcoding Publisher storage facility that allows fragment retrieval. The resource repository allows reuse by other components needing a similar service in the future. The fragmentor sends the first generated fragment to the device and stores the remaining generated fragments in the resource repository.

Content Based Routing (CBR)

The Content Based Routing (CBR) module of Load Balancer works with Caching Proxy to proxy client requests to specified servers. Caching Proxy allows for faster document retrieval with low network bandwidth requirements. With CBR, you may specify a set of servers to handle requests based on regular expression matching of the content of the request. Since multiple servers handle each type of request, the requests can be load balanced for optimal client response. CBR also detects when one server in a set has failed and stop routing future requests to that server. Load Balancer's CBR module can be installed on the same machine as the Caching Proxy. Also, the machines to which requests are being routed do not have to be on the same LAN segment as the CBR machine.

Cookie affinity

Load Balancer's CBR function supports cookie affinity. The cookie affinity feature applies only to CBR with Caching Proxy, which supports load balancing based on rules. This combination of components provides a new way to make clients "sticky" to a particular server. With cookie affinity enabled, the server that first serviced an end user's request is recorded in a special packet of data (a cookie) included in the server's response. When the end user accesses the same URL again within a period of time that you define, and the request includes the cookie, CBR routes the request to the original server rather than reapplying its standard rules.

Once a rule has been enabled for cookie affinity, new client requests are load-balanced using standard CBR algorithms while succeeding requests from the same client are sent to the initially chosen server. The chosen server is stored as a cookie in the response to the client. The cookie is then inserted in the headers that go back to the client, and if the client's browser is configured to accept cookies, it sends back subsequent requests. As long as the client's future requests contains the cookie, and each request arrives within the stickytime interval, the client maintains affinity with the initial server.

The following demonstrates one way in which the process might work.

1. A WAP phone requests a URL
2. The Load Balancer, CBR+Cookie affinity dispatches the request to Transcoding Publisher 1
3. Transcoding Publisher 1 serves the request, fragments the resource, and sends the first fragment to the microbrowser
4. The WAP phone requests the second fragment
5. CBR+Cookie affinity routes the request back to Transcoding Publisher 1
6. Transcoding Publisher 1 retrieves required fragment if is still cached.

How to enable cookie affinity with the `rule set` command

- `rule set cluster:port:rule stickytime 60`
- `rule set cluster:port:rule affinity cookie`
- `rule set cluster:port:rule stickytime 300`
- `rule set cluster:port:rule affinity clientip`

Stickiness is set per rule. The default for rule affinity is the client IP, so if you haven't set it to cookie, you do not need to set it to `clientip`.

Considerations when running Transcoding Publisher behind WebSEAL-Lite

When creating deck fragments for a large WML document, Transcoding Publisher encodes a unique identifier in a special anchor tag in each fragment that is used to navigate between fragments.

When running Transcoding Publisher behind another proxy server that is configured to run as a reverse proxy such as WebSEAL-Lite, the deck navigation links that Transcoding Publisher inserts are not aware of the fact that Transcoding Publisher is behind a reverse proxy server. Because of this, the requests are not sent correctly, and the client is unable to reach subsequent fragments.

To make this scenario work, two things have to be done:

1. Transcoding Publisher must output the address of the WebSEAL-Lite host as part of the unique fragment identifier. Change the values from `ifrag-` to `webSEAL-Lite.server.name:80/ifrag-` as follows:

   From:
   ```
   /etc/plugins/ibm/FragmentationEngine/FragmentationEngineConfiguration/FragmentSpecifier
   = ifrag-
   ```
   To:
   ```
   /etc/plugins/IBM/FragmentationEngine/FragmentationEngineConfiguration/FragmentSpecifier
   = webSEAL-Lite..server.name:80/ifrag-
   ```

   From:
   ```
   /etc/plugins/IBM/FragmentationEngine/FragmentationEngineConfiguration/PrimarySpecifier
   = ifragp-
   ```
   To:
   ```
   /etc/plugins/IBM/FragmentationEngine/FragmentationEngineConfiguration/PrimarySpecifier
   = webSEAL-Lite.server.name:80/ifragp-
   ```
2. If you are using proxy directives to configure a reverse proxy, the Caching Proxy (Web Traffic Express) hosting WebSEAL-Lite must be configured to not remove its address from the URI that it passes on to Transcoding Publisher. Add a directive to the Caching Proxy ibmproxy.conf file to match the URL that Transcoding Publisher is expecting:

   ```
   Proxy /ifrag* http://wtp.server.name:8081/ifrag*
   ```

Note: The fragment specifier is an example and can be set to any valid values.

Related information

- WebSphere Everyplace Server authentication and security

Working with Voice Services

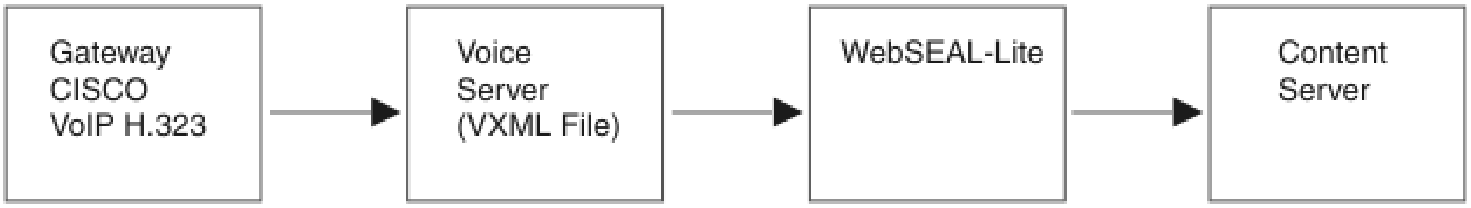
Voice Services for WebSphere Everyplace Server 2.1 allows telephone uses to authenticate their identity to access voice enabled application located in a protected part of the network. Voice Services is a voice version of the authentication process for web browser authentication.

When accessing a secure website via a browser, the user is asked to enter their ID and password through an HTML form. The server determines if they have the appropriate access and allows or denies access to the site.

When a user accesses a secure application through Voice Services, the user calls a subscriber phone number. This results in a Voice XML(VXML) Browser running on the Voice Server handling the incoming call. The IBM VXML browser prefetches the page before it handles a call.

Once a call is detected and answered, the browser starts to process the login VXML. The user is then prompted for a numeric userid and numeric password to authenticate their identity to WebSphere Everyplace Server. The user can either speak the numbers or use DTMF tones; the recognition grammar supports both. The resulting form information is posted back to WebSEAL-Lite. The information is then passed to the forms based plug in and is authenticated.

If the login is successful, session cookies are returned to the Voice Server for future requests to prove the user has been authenticated. The user is then directed to the target page that was requested with the login form. If the login fails, WebSEAL-Lite returns an VXML file to inform the user of the failed login and then hangs up.



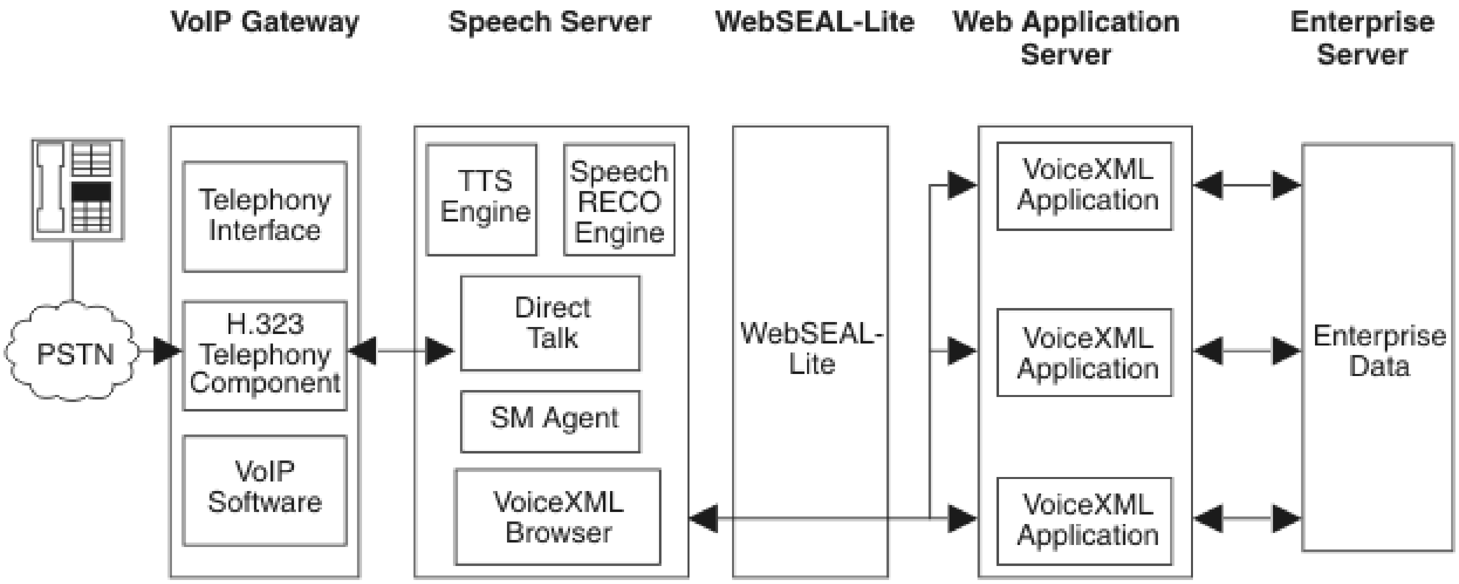
Voice Services is compliant with VXML 1.0 Voice Services. However, we have only tested and verified compliance with the AIX server and Motorola Server.

What are the components of Voice Services?

To run Voice Services, there are three hardware/software components which must be installed and set up correctly prior to setting up Voice Services.

- **Via Voice -** Via Voice delivers low cost, very low resource Speech Recognition. Via Voice provides command and control, RTOS Support, speaker dependent language independent, speaker independent language independent, and a 8 kHz sampling rate. In Voice Services, Via Voice translates the users speech when Voice Services is authenticating the user. The text is then used by the VXML file to determine the user is authorized.
- **Direct Talk -** Direct Talk allows enterprises to deploy voice-enabled applications on a traditional IVR network infrastructure. Direct Talk uses the existing DT platform and IVR network infrastructure, enables current Direct Talk users to transition to a Web-development paradigm using Voice XML, and supports the Voice XML 1.0 specification. In Voice Services, Direct Talk translates the VXML file from text to speech. This allows the user to hear the speech version of the VXML file.
- **Voice Server -** Voice Server allows you to deploy web-based Voice applications written in VXML. In Voice Services, the Voice Server is the server where the WebSphere Everyplace

Server Voice Services application and VXML file is installed.



Related information

- Configuring Voice Services
- Configuring WebSEAL-Lite
- Personalizing Voice Services
- Provisioning Support for Voice Services

Installing Voice Services

- Hardware/Software Prerequisites
- Installation

Hardware/Software Prerequisites

Voice Services is installed during the WebSphere Everyplace Suite installation. Before configuring Voice Services, you must have the following hardware/software installed:

- AIX
- ViaVoice
- IBM WebSphere Voice Server for Direct Talk
- Direct Talk 6000 R 1.0

Follow the installation instructions shipped with the hardware/software.

Installation

During the WebSphere Everyplace Server installation, select to install the WebSphere Edge Server - Caching Proxy component. In the subcomponent list on the right side of the dialog box, select WebSEAL-Lite. Voice Services is installed during the WebSEAL-Lite installation.

Related information

- Configuring Voice Services
- Configuring WebSEAL-Lite
- Personalizing Voice Services
- Provisioning Support for Voice Services

Personalizing Voice Services

This section describes customizing the Voice Services VXML files to personalize Voice Services. There are three files you need to customize which are located in the Samples directory under WebSEAL-Lite.

- Customizing wesloginform.vxml
- Customizing wesloginfail.vxml
- Customizing weshelloworld.vxml

Customizing wesloginform.vxml

The *wesloginform.vxml* file controls the message the user hears when the page displayed to user when the user is either authenticated or login denied. Use this file to customize the attributes of Voice Services particular to your company and the applications accessed by the user. This file is located in the directory where Voice Services is installed

1. Open the *wesloginform.vxml* file in a text editor.
2. Enter the target URI for the portal page you want displayed. The sample includes the sample portal included in Voice Services.

   ```
   <var name="targetURI" exp="'<yourtargetURI>'"/>
   ```

   where <your target URI> is the URI of the portal page you want displayed for the login
3. Customize any additional variables you want to personalize for Voice Services.
4. Save and close the file.

Customizing wesloginfail.vxml

The *wesloginfail.vxml* file is a sample VXML file which shows how the VXML Voice Server can authenticate a userid and password. Use this file to customize the page displayed when the user's authentication fails.

1. Open the *wesloginfail.vxml* file in a text editor.
2. Enter the target URI for the web page you want displayed when the user's authentication fails. The sample includes the sample web page included in Voice Services.

   ```
   <var name="targetURI" expr="'<your target URI>'"/>
   ```

   where <your target URI> is the URI of the portal page you want displayed where the user is not authenticated through Voice Services
3. Customize any additional variables you want to personalize for Voice Services.

4. Save and close the file.

Customizing weshelloworld.vxml

The *weshelloworld.vxml* file is one of the sample files which shows how a VXML voice server can authenticate a userid and password. Use this file to customize where the login page is located.

1. Open the weshellowworld.vxml file in a text editor.
2. Enter the target VXML file you want to use to determine the login form that is displayed.

   ```
   <var name="targetURI" expr="'<LoginVXMLFile>'"
   ```

   where <targetURI> is the VXML file that denotes the login page
3. Customize any additional variables you want to personalize for Voice Services.
4. Save and close the file.

Related information

- Working with Voice Services
- Configuring Voice Services
- Configuring WebSEAL-Lite
- Provisioning Support for Voice Services

Provisioning Support for Voice Services

Each Voice Services user must have voice preferences set up which is information necessary for the user to use the telephone interface. One of the most important voice preferences is the users telephone number. This number is used as the user ID for authentication purposes. The administrator must originally set up the user using Tivoli Service Manager or the user can use Self-Care to set themselves up. After the initial setup, the user can use the Self-Care screens for updating their own information. The Self-Care screens should be configured as a set of protected resources by the administrator.

Self-Care

Because these steps are performed by the Voice Services user, provide these instructions to your users. Use Self-Care to perform the following tasks.

- Logging into Self-Care
- Changing Voice Services Password
- Changing Personal Information

Logging into Self-Care

1. Access the Voice Services Self-Care Login Screen in Tivoli Service Manager. To access Voice Services Preferences, you must be authenticated through WebSphere Everyplace Server. The Tivoli Service Manager provides you with the URL for this screen.
2. If prompted to login on the Self-Care Login Screen, enter your User Name and Password and click the OK button. The Self-Care session is activated.

Changing Voice Services Password

1. After you are logged into the Self-Care session, click Password. The Password screen is displayed.
2. Enter the current password.
3. Enter the new password. Note that you must enter a numeric password that can be entered through a mobile phone.
4. Retype the new password and click Save.

Setting Telephony Preferences

1. After you are logged into the self-care session, click the Change Telephony

Preferences link. The Change Telephony Preferences screen is displayed.

2. Enter the telephone number that is going to be the alias for the user id.
3. Click Submit.

Changing Personal Information

1. After you are logged into the self-care session, click the Personal Information link. The Personal Information screen is displayed.
2. Modify the information you want to change and click Save. The changed information is updated.

Related information

- Working with Voice Services
- Installing Voice Services
- Configuring WebSEAL-Lite
- Personalizing Voice Services

Configuring Voice Services

This section describes configuring Voice Services and contains the following sections:

- Setting up VXML file
- Setting up the Voice Server to Access Resources Through WebSphere Everyplace Server
- Configuring LDAP
- Related information

Setting up VXML file

The VXML file controls the speech heard by the user when accessing Voice Services. The VXML file is installed during the WebSphere Everyplace Server installation and placed in the Sample directory. You need to put the VXML file onto a secure server.

1. Copy the *wesloginForm.vxml* file.
2. Place the file on a secure HTTP Server, such as a Voice Server.

Setting up the Voice Server to Access Resources Through WebSphere Everyplace Server

You need to configure the Voice Server to request the first page of VXML supporting the user application the user is attempting to access. The configuration file controls the request to VXML. This file contains parameters you can customize based on how you want to use Voice Services.

1. Verify all software required to run Voice Services is installed and set up correctly.
2. Open the configuration file in a text editor. The file is located in the directory where you installed Voice Services.
3. Modify the parameters you want to customize. The following table details the parameters in the file.

Property	Description

NUMBER_INBOUND_BROWSERS	Specifies the number of inbound Voice SML browsers the system starts for incoming calls. One set of inbound property variables must be defined for each specified browser. Pre-configured value=1 Required Valid values - decimal integers>0
INBOUND_BARGE_IN*n*	Indicates the barge-in detection method used by the Voice XML browser. Default = recognition Valid values =energy or recognition
INBOUND_DUPLEX*n*	Specifies the duplex implementation. When full-duplex is specified, the user and computer can speak at the same time when half-duplex is specified, the user should not speak while the computer is speaking because the audio is not received by the speech recognition engine. Default = full Valid values=full or half
INBOUND_H323_ENDPOINT_ALIAS*n*	An H.323 terminal endpoint identification used for routing calls from the VoiP gateway through a VoIP gatekeeper to an IP network. Note: The alias must be a telephone number based on the E164 standard. Valid values = any numeric values **With a VoiP gatekeeper:** Default=none Required Numeric - (all digits) **Without a VoIP gatekeeper:** Default = 9999

INBOUND_SITE*n*	Specifies the URL of a VoiceXML site customization file that is read when the Voice XML browser starts up. The file remains loaded for the life of the VoiceXML browser instance. Default=none Valid values=existing URL
INBOUND_TIMEOUT*n*	Specifies the timeout value in seconds before the VoiceXML browser throws a no input event, in the form of numbers. Default = 7 Valid value = decimal integers>=2

4. Save and close the file.

Configuring LDAP

You need to configure LDAP to define the Voice Server as the gateway for Voice Services. This tells WebSEAL-Lite to use the phone number for the Voice Services to look up the users ID and password. Otherwise, WebSEAL-Lite does not know that the user ID is aliased through the phone number and does a straight lookup. Because the user is not registered this way, WebSEAL-Lite does not recognize the user ID and password and the user is not be authenticated.

1. Open the Authentication Server Configuration file *ibmwesas.conf*. It is typically located in /usr/lpp/IBMEPS.
2. For each gateway, add a gateway DN statement to your Authentication Server Configuration file. This statement specifies the gateway's location in the LDAP tree, relative to the baseDN. The syntax for this statement is:

   ```
   gatewayDN DN_relative_to_baseDNh
   ```

   ```
   For example: gatewayDN cn=nokia
   ```
3. In LDAP, open the Directory Management Tool. You can use this tool to add entries to your LDAP tree describing each of your third-party gateways.
4. Add the main container.

   . Highlight the location in your LDAP tree for each type of third-party gateway you plan to use. This location must be under the based DN (the LDAP Distinguished Name) specified in your Authentication Server configuration file, but not under the cn=root, sys=ewg entry. This is because this directory is reserved for use by IBM's Everyplace Wireless Gateway.

   b. To add a container, click the Add button. The Add an LDAP Entry dialog box is displayed.

c. In the Entry RDN field, enter "cn=<gateway_name>". The gateway name must be the same name you entered in the configuration file. For example, if you entered nokia in the configuration file, enter "cn=nokia".

d. In the Structure Object Class, select Container.

e. Click the OK button. The Add an LDAP Entry prompt is displayed with the values for the container. Verify the values.

f. Select the Add button. The LDAP tree is displayed with the new container.

5. Add a sub-container for the common entry.

   . Highlight the container you just created.

   b. Click the Add button. The Add an LDAP Entry dialog box is displayed.

   c. In the Entry RDN field, enter "cid=Common".

   d. In the Entry Type field, select Other. Two additional tabs are displayed.

   e. In the Structure Object Class, select eProperty Set.

   f. Click the OK button. The Add an LDAP Entry prompt is displayed with values for the container. Verify the values.

   g. Click the Add button. The LDAP tree is displayed with the new entry.

6. Create a sub-container for the ClientIDAttributeName.

   . In the LDAP tree, highlight cid=common.

   b. Click the Add button. The Add an LDAP Entry dialog box is displayed.

   c. In the Entry RDN field, enter "settingID=clientIDAttributeName".

   d. In the Entry Type field, select Other. Two additional tabs are displayed.

   e. In the Structure Object Class, select eProperty.

   f. Click the OK button. The Add an LDAP Entry prompt is displayed with values for the container. Verify the values.

   g. In the CisProperty field, enter "<mobile>". This represents the mobile phone number for a each user.

7. Create a sub-container for the plug in name.

   . Highlight "cn=<gateway_name>".

   b. Click the Add button. The Add an LDAP Entry dialog box is displayed.

   c. In the Entry RDN field, enter "settingID=pluginName".

   d. In the Entry Type field, select Other. Two additional tabs are displayed.

   e. In the Structure Object Class, select eProperty.

   f. Click the OK button. The Add an LDAP Entry prompt is displayed with values for the container. Verify the values.

   g. In the CesProperty field, enter "<mobile>". This represents the mobile phone number for a each user.

8. Create a sub-container for the Voice Server name.

   . Highlight "cn=<gateway_name>".

b. Click the Add button. The Add an LDAP Entry dialog box is displayed.
c. In the Entry RDN field, enter "cn=voice_server_name." This is the name of the actual voice server you are using with the WebSphere Everyplace Server Voice Services.
d. In the Entry Type field, select Other.
e. In the Structure Object Class, enter "w1Gateway".
f. In the Auxiliary Object Class field, enter "ibm-w1Resource".
g. Click the OK button. The Add an LDAP Entry prompt is displayed with values for the container. Verify the values.
h. In the Host field, enter <IP_Address_for_Host>.

9. Restart Web Traffic Express. If you installed the debug version of *wesauth.so*, you can set the debug message level via the "debudLevel" statement in the WebSEAL-Lite configuration file to provide the debugging/progress information in one of the Web Traffic Express log files.

Related information

- Working with Voice Services
- Configuring WebSEAL-Lite
- Personalizing Voice Services
- Provisioning Support for Voice Services

Configuring WebSEAL-Lite

WebSEAL-Lite is the authentication server that authenticates users for Voice Services. When a user enters login information, the Voice Server sends the authentication information to WebSEAL-Lite for authentication. You need to configure the Authentication Server to recognize that the location receiving the form based login is the telephony gateway. The Authentication Server uses the user ID and password to look up the user information in the LDAP store.

You need to configure two files:

- *osdef.conf*
- *ibmwesas.conf*

Working with Osdef.conf

This file is the Object Space Definition File and defines the mapping between URLs and the Policy Director object space. Using this file, you can define which branch of the object space WebSEAL-Lite should use to perform authorization checks against users based on the domain name in the URL. You can also use this file to specify domain specific configuration options for WebSEAL-Lite. For Voice Services, you need to modify the Global settings section. All options in the global settings section are applied to all objects in the object space that do not explicitly override the corresponding global setting.

1. Open the WebSEAL-Lite Configuration file *osdef.conf*.
2. You need to specify the login method for the domain as forms. This tells the Voice Server to use a form based login for the login method. You can also associate the login method with a specific device profile. If no device is specified, all devices use the same login method. Modify the following statement.

   ```
   <login_method>= <device profile lists>
   ```

   For example

   ```
   forms=nokia_device_palm_device
   ```

3. You need to specify the form error file to be displayed to the user when the authentication form fails. You can use the same file as the login form and add an error message indicating that the login failed. Modify the following statement:

   ```
   form_login_errorfile=location of WES Login Failure VXML
   ```

4. You need to specify the file to be displayed when the user logs out. This file also cleans up a cookie on an authentication server. WebSEAL-Lite deletes the user's session information when the user makes a request matching the URL. Modify the following statement:

   ```
   <local path | URL> device profile list
   ```

For example:

```
form_logout_file =/pub/forms/logout_pm.html nokia device
```

5. You need to specify a form signature for WebSEAL-Lite. A form signature is a hidden attribute to value assignment in a form. If WebSEAL-Lite receives a form submission with this assignment when form login is the login method, it extracts the userid and password from the form to authenticate the user. This allows WebSEAL-Lite to authenticate a user even if the user had not previously tried to access a protected web page. If this option is not specified, no form login signature is checked. Uncomment the following statement by removing the # sign in front of the statement:

   ```
   Form_signature_login=FormType=Login Form
   ```

6. You need to specify the user login field name. This is the field name of the User ID submitted to WebSEAL-Lite during the authentication process. The default is User ID. Uncomment the following statement by removing the # sign in front of the statement and then enter the specific user ID you want to use.

   ```
   Form_fieldname_userid=UserID
   ```

7. You need to specify the user password. This is the field name of the User ID submitted to WebSEAL-Lite during the authentication process. The default is Password. Uncomment the following statement and enter the password you want to use:

   ```
   Form_fieldname_password=Password
   ```

Working with ibmwesas.conf

This file initializes and starts WebSEAL-Lite. Use this file to set initialization options such as authentication method, Policy Director initialization parameters, and the LTPA module. You can also use the object space and user mapping configuration files to set additional configuration options.

1. Open the *ibmwesas.conf* file.
2. Generate the LTPA cookie key file.
3. Uncomment out the following section:

   ```
   LTPA_Cookie_Enabled     Yes
   LTPA_Cookie Keyfile /opt/pdweb-lit/conf/ltpa.keys
   LTPA_Cookie_Keyfile_Password    secret 5
   LTPA_Cookie_TTL
   LTPA_Cookie_Contains_Password   No
   ```

4. Comment the following line by placing a # in front of the line.

   ```
   LTPA_Cookie_Enabled No
   ```

5. In the LTPA_Cookie_Keyfile line, enter the location of the keyfile you generated.
6. In the LTPA_Cookie_Password line, enter the password for the keyfile.

Related information

- Working with Voice Services
- Configuring Voice Services
- Personalizing Voice Services
- Provisioning Support for Voice Services

User Preferences

Everyplace Server User Preferences collects data about users and their devices. User Preferences provides one common place to collect and manage user data for Everyplace Server, including preferences for Intelligent Notification Services, Location Based Services, and Voice Services, if the associated components are installed on the WebSphere Everyplace Server Domain. This section includes information on modifying the User Preferences interface as well as the SecureWay Directory structure.

Overview

User Preferences interface

User Preferences builds on the Tivoli Internet Services Manager enrollment, self-care, and customer-care preferences interface by providing administrators with customizable JSP samples that allow the end user to enter information with a browser. The User Preferences interface runs on the Application Server and is accessible from the Tivoli Internet Services Manager menu. The JSP samples provided can be adapted to display specific content. Data collected through the user entry interface is mirrored in the SecureWay Directory.

User Preferences design

Customizing the SecureWay Directory schema in conjunction with the JSP templates allows for more flexibility in the data collected during user entry. Each attribute of the user entry structure holds specific values that can be modified with various tools, including SecureWay Directory Manager and Tivoli Internet Services Manager.

Installation

See the Everyplace Server readme for instructions on installing User preferences.

Migration

User Preferences interface

Existing Everyplace Server customers must manually migrate to the new User Preferences items. Follow the installation instructions above to install the preference samples to the proper directory. Be sure to migrate any SecureWay Directory or JSP customizations made in prior versions to the new set of web applications before deploying the pages.

User Preferences design

When Setup Manager is run at a server where SecureWay Directory is installed, Setup Manager detects the existing Everyplace Server Directory Information Tree (DIT) version number and prompts with a message indicating that the DIT must be migrated to the current release. Setup Manager offers you an option to continue the migration to the current Everyplace Server version DIT or cancel the installation.

If you install other Everyplace Server components and SecureWay Directory information is available, Setup Manager detects the existing DIT version number. If the version number is not current, Setup Manager prompts you with a message indicating that the DIT must be migrated to the current version. Setup Manager then closes.

User Preferences interface tasks

Three levels of user access define data entry to the User Preferences JSPs. In addition to Administrators or Customer Developers, subscribers and Customer Service Representatives (CSR) may add or modify directory information by accessing deployed JSPs with a browser. The administrator may modify all pages delivered to the end user. Each user level limits the type of information entered by the respondent.

Self-care
A subscriber is able to enroll; change passwords and secret data items; view account information; change billing and payment options; add, change, or cancel content or subaccounts. A typical user may perform the following tasks when accessing his or her account:

- Access User Preference pages via browser
- Define notification preferences, including groups, devices, and profile info
- Define location, voice, and synchronization preferences

Customer Service
The creation of customer care JSPs require the modification of existing self-care JSP templates. A CSR has access to the same information available to the end user. The CSR can also search for current subscribers, apply customer payments, view devices, and create reports. A typical CSR performs tasks similar to the end user when accessing a customer account.

Customer Developer/Administrator
Customer Developers or Administrators are able to perform the same tasks as self-care and CSR users. However, an administrator can also modify pages delivered to the end user before deploying the JSPs on the Application Server. All modifications should be supported by the SecureWay Directory schema. The administrator may perform the following tasks to prepare for access by the end user or CSR:

- Migrate prior JSP updates
- Customize the look, feel, and content of pages for end users
- Supply customer care JSPs based on sample provided by self-care JSPs
- Deploy JSPs for access
- Modify user information in the SecureWay Directory through the JSPs

Customization

User Preferences interface - JSP templates
The JSPs that shape the User Preferences interface can be modified with a company's logo, colors, and specialized content. Be sure that the SecureWay Directory schema reflects any data collection changes made in the User Preferences interface.

User Preferences design - SecureWay Directory schema
All changes to the User Preferences design correspond to SecureWay Directory and can include modifications to object classes, attributes, and values as well as data about specific users, groups, and devices. Administrators can modify these properties with Everyplace Server's Directory Management Tool, Tivoli Internet Services Manager, or a ASCII text editor. Creating and altering an LDIF is also a suitable way to apply changes to the directory on a large scale. Modifications to the SecureWay Directory schema should be approached with caution. Previous LDAP experience is recommended.

The User Preferences structure is detailed in the following table. Click on a link to view additional information on the attributes associated with an object class and the values they take.

Directory Information Tree - Architecture of a Realm Subtree	
Realm - `organizationalUnit`	
Users - `container`	
User - `inetOrgPerson`+`ePerson`	

Service Attributes per User	
ibm-undUser ibm-insUser secUser	Inherited Object Classes top cimManagedElement eUser
User's Devices	
ibm-device ibm-undDevice	Inherited Object Classes top cimManagedElement cimManagedSystemElement cimLogicalElement cimLogicalDevice
User-Defined Groups	
groupOfNames ibm-undGroup	Inherited Object Classes top

ibm-SdpUser+wlUser

```
top
person
organizationalPerson
inetOrgPerson
ePerson
ibm-CertificateForDN
ibm-deviceList
cimManagedElement
eUser
ibm-SdpUser
```

ibm-CertificateForDN

```
top
person
organizationalPerson
inetOrgPerson
ePerson
ibm-CertificateForDN
```

ibm-deviceList

```
top
person
organizationalPerson
inetOrgPerson
ePerson
ibm-CertificateForDN
ibm-deviceList
```

An object class is just a mechanism for defining a collection of attributes for the instatiation of a directory entry, so you may use or reuse an attribute from an object class that you don't need. Be sure to examine all the attributes in the existing SecureWay Directory schema. You may still need to define new objects and attributes to put your information into the SecureWay Directory. Subclass the objects where possible and define new objects only when the current ones aren't adequate for your needs.

Each person entered into Tivoli Internet Services Manager creates a person entry in the SecureWay Directory for each subscriber. A person entry consists of the object classes listed in the ibm-SpdUser+ePerson object class.

The following table indicates which attributes and values exist for each object class. Sample coding values and format follow each object class. The attributes listed below are required unless otherwise noted.

Object Class/Description	Existing Attributes	Additional Comments
organizationalUnit Realms contain realm nodes, defined in Tivoli Personalized Services Manager.	o=<organization>	To formulate the DN for a realm, parse the realm name from the HTTP header inserted by WebSEAL-Lite for inbound requests.
	ou=<organizational unit>	
	Example:	
	dn: ou=isp,dc=austin,dc=ibm,dc=com	
container Holds a user entry for each subscriber in the realm.	cn=<container name>	Containers may include user entries that contain specific attributes associated with a user.
	Example:	
	dn: cn=users,ou=isp,dc=austin,dc=ibm,dc=com dn: cn=insUser,uid=jdoe,cn=users,ou=isp,dc=austin,dc=ibm,dc=com	

inetOrgPerson Enables services to depend on a standard object class.	--	LDAP standard object class.
ePerson Everyplace Server can depend on a set of attributes for each person entry regardless of the structural object class attached.	uid=<username>	An auxiliary object class. Default structural class used is inetOrgPerson.
	userPassword=<password>	
	Example:	

	uid=jdoe
	userPassword=mypwd	

ibm-SdpUser Identify the Everyplace User as running a client.	ibm-WapClient: <t/f>	
	ibm-WgClient: <t/f>	
	ibm-tismStatus	I=Initial, active account. C=Connected, a fully enabled account. D=Disconnected, an inactive account. L=Logically Deleted.
	Example:	
	objectclass: ibm-SdpUser ibm-WapClient: TRUE ibm-WgClient: TRUE ibm-tismStatus: C	
wlUser	--	LDAP standard object class.
ibmCertificateForDN Contains certificate and issuer DN combinations outside user certificate or other DER-encoded certificate value.	ibm-CertificateSubjectAndIssuer:	Contains extracted values of certificate in XML string. Optional.
	Example:	
	ibm-CertificateSubjectAndIssuer: <certDN>cn=cert</certDN><issuerDN>cn=issuer</issuerDN>	
ibm-deviceList Lists devices assigned to the user.	ibm-deviceIDList=<device IDs>	
	Example:	
	ibm-deviceIDList: 5555552155 ibm-deviceIDList: 5555556378	
ibm-undUser Sets user attributes specific to the Universal Notification Dispatcher	c>n=<container name>	
	ibm-deviceIDList=<device IDs>	
	ibm-groupList=<user group list>	
	Example:	
	objectclass: ibm-undUser cn: undUser ibm-deviceIDList: 5555552155 ibm-groupList: cn=group3,uid=jdoe,cn=users,ou=isp,dc=austin,dc=ibm,dc=com	
ibm-insUser Sets user attributes specific to Intelligent Notification Service.	cn=<container name>	
	ibm-insSubscriptionType	Type of subscription.
	ibm-insMaxContentPersistTimeInSeconds	Maximum persist time.
	ibm-insMaxContentStorageSizeInKB	Maximum storage per user.
	ibm-insTriggerFiringFrequency	Determines events after firing.
	ibm-insAuthorizedContentSources	List content sources.
	Example:	

	cn: insUser ibm-insSubscriptionType: SAVE ibm-insMaxContentPersistTimeInSeconds: 36000 ibm-insMaxContentStorageSizeInKB: 1024 ibm-insTriggerFiringFrequency: ONCE ibm-insAuthorizedContentSources: sports ibm-insAuthorizedContentSources: stocks	
secUser Sets user attributes specific to Policy Director.	--	--
ibm-device Holds attributes for each device assigned to a subscriber.	deviceID	
	ibm-deviceIDType	1=MSISDN, 2=MIN, 3=IP address.
	ibm-isDeviceEnabled: <t/f>	
	description	Optional.
	ibm-deviceType	1=Pager, 2=Fax, 3=Voice, 4=SMS, 5=WAP. Optional.
	ibm-deviceOS	Optional.
	Example:	
	objectclass: ibm-device deviceID: 5555552155 ibm-deviceIDType: 1 ibm-isDeviceEnabled: TRUE ibm-deviceOS: PalmOS 3.0 ibm-deviceType: 5	
ibm-undDevice Holds additional attributes required by UND to send notifications to a user using a specific device.	cn=<container name>	Unique name for each entry.
	host	Host name of server or gateway communicating with device.
	ipServicePort	Port number for device.
	ibm-appDeviceAddress	Device address used to communicate with device.
	ibm-appProtocol	Protocol used to communicate with the device.
	ibm-appProtocolVersion	Protocol version used to communicate with the device.
	ibm-appProtocolType	Type of device communication.
	uid	User id to access device. Optional.
	userPassword	User password to access device. Optional.
	description	Optional.
	Example:	
	objectclass: ibm-undDevice cn: undDevice host: myhost.austin.ibm.com ibm-appDeviceAddress: 9.153.4.6 ibm-appProtocol: sametime ibm-appProtocolType: im ibm-appProtocolVersion: 1.5 ipServicePort: 1000 uid: jdoe userPassword: mypwd	
groupOfNames	cn=<group name>	

	description=<user defined group>	Optional
	member=<DNs>	
	Example:	
	objectclass: groupOfNames cn: mygroup member: uid=member1,cn=users,ou=isp,dc=austin,dc=ibm,dc=com member: uid=member2,cn=users,ou=isp,dc=austin,dc=ibm,dc=com member: uid=member3,cn=users,ou=isp,dc=austin,dc=ibm,dc=com	
ibm-undGroup Per user attributes for notification dispatcher.	cn=	
	ibm-undAuthorizations	Group members authorized to send message by using UND.
	description	
	ibm-undUrgentDevicesAuthorization	Lists deviceIDs for urgent message devices. Optional.
	ibm-undNormalDevicesAuthorization	Lists deviceIDs for normal normal message devices. Optional.
	ibm-undFYIDevicesAuthorizations	Lists deviceIDs for FYI messages. Optional.
	Example:	
	objectclass: ibm-undGroup cn: undGroup ibm-undAuthorizations: 567:0X0F06EF64 ibm-undUrgentDevicesAuthorizations: 5555556378 ibm-undNormalDevicesAuthorizations: 5555556378 ibm-undNormalDevicesAuthorizations: 5555552155 ibm-undFYIDevicesAuthorizations: 5555552155	

Component attributes

In addition to the base enrollment preferences collected by Tivoli Internet Services Manager, self-care, CSRs, or Administrators can access the User Preferences JSP template and modify user settings for the following Everyplace Server components with the User Preferences.

Intelligent Notification Services
Intelligent Notification Services use Short Message Service messages, Wireless Application Protocol push messages, e-mails, and instant messaging to notify users when preconfigured events occur. Users access the JSP templates and enter their preferences about the delivery method and type of events sent to them. Attributes related to this component include `ibm-deviceList`, `ibm-scsUser`, `ibm-undUser`, `ibm-insUser`, and `secUser`. Intelligent Notification Services includes details about the user entry interface and how to interact with it.

Location Based Services
Location Based Services allow the transmittal of messages that are specific to a user's geographical location. Users define a list of URLs and create privacy preferences that permit or restrict each listed web site from viewing location information. Attributes related to this component include `eProperty` and `ePropertySet`. More information on working with Location Based Services is available in the Location Based Services section.

Voice Services
Voice Services allows a user to set up an alternate alias for telephony access. The alias typically consists of a numeric entry (a telephone) number to substitute for the user id when connecting through this method. Once the user successfully establishes a numeric alias, Voice Services prompts the user to change his or her existing password to a numeric entry. The affected object class and attribute are `mobileTelephoneNumber/ePerson ormobile/ePerson`.

User Device Identifiers
Only customer service representatives may enter user device identifiers in the User Preferences templates when

enrolling or updating customers. This information provides user overrides for device characteristics by altering `ePropertySet` + `eProperty` under the person object.

Language Services
The Language Services section of User Preferences allows the end user to select a language preference for content when a transmission is capable of alternate language delivery. The appropriate object class for this preference is `preferredLanguage/ePerson`.

User Certificate DN
This User Preferences attribute allows the user to set the DN from his X.509 certificate (if that user is doing certificate authentication) and allows Everyplace Server to map from that certificate DN back to the original user SecureWay Directory entry. The affected object class is `secCertDN/ibm-SdpUser`.

Related information

- Intelligent Notification System Preferences
- External information

Change SecureWay Directory passwords for Everyplace Server components

SecureWay Directory passwords set during installation

Two different sets of SecureWay Directory user IDs and passwords are defined during Everyplace Server installation.

- A user ID and password to administer SecureWay Directory.
- A user ID and password that each component uses to communicate with SecureWay Directory.

Both sets of user IDs and passwords are defined during Everyplace Server installation. The Everyplace Server component password can either be the same or different for each Everyplace Server component that accesses SecureWay Directory.

Changing SecureWay Directory passwords for Everyplace Server components

SecureWay Directory passwords for Everyplace Server components can be changed by command line or by using Everyplace Suite Manager.

Change the SecureWay Directory password (the SecureWay Directory user ID cannot be changed) using the following command-line statement:

For AIX:

```
cd /usr/lpp/IBMEPS.Inst
./ChangePassword.sh component userid current password
new password confirm new password
```

For Solaris:

```
cd /opt/IBMEPSIn
./ChangePassword.sh component userid current password
new password confirm new password
```

There are two ways to use the command line interface.

1. Enter all the parameters at once:

   For example, type the following to change the SecureWay Directory password for:

   - WebSphere Transcoding Publisher

     ```
     ./ChangePassword.sh Transcoding old_uid old_password new_password new_password
     ```

   - Tivoli Personalized Services Manager

     ```
     ./ChangePassword.sh Tivoli old_uid old_password new_password new_password
     ```

2. Enter the parameters one at a time:

   For example, type the following to start:

   ```
   ./ChangePassword.sh
   ```

   and press **Enter**. You are prompted for each of the above parameters in turn.

Use the corresponding keywords to change the password for the following components:

Component	Component keyword
Everyplace Active Session Table	ActiveSession
Everyplace Intelligent Notification Services	Notification
Everyplace Location Based Services	Location
Everyplace Server uninstall program	uninstall
Everyplace Suite Manager	Console
Everyplace Wireless Gateway	Gateway
Everyplace Wireless Gateway Gatekeeper	Gatekeeper
Tivoli Personalized Services Manager	Tivoli
WebSEAL-Lite	Proxy
WebSphere Transcoding Publisher	Transcoding

Note: The SecureWay Directory passwords can also be changed using a Web browser. Instructions about doing this can be found in the SecureWay Directory documentation.

After you change the password

After you change the password at the SecureWay Directory server, you must also change this password for all of the other servers in the Everyplace Server domain, or you are not able to access SecureWay Directory from those servers.

You can change separate component administrator passwords through component administration consoles by following instructions provided for that component.

Component processes

Some WebSphere Everyplace Server components may require starting and stopping after installation. Starting and stopping various components may also be necessary after additional component installation, migration, or reconfiguration. This section details the start and stop processes as well as other pertinent information regarding component configuration. Refer to the component documentation for further configuration and administration information.

In most cases components can be started and stopped from the command line or from Everyplace Suite Manager. See Starting Everyplace Suite Manager for more information.

Unless otherwise indicated, you must be in the component's product_installation_root/ directory to execute command lines. You may also access some components with Suite Manager while other components may have their own administration consoles.

DB2 Listener	

Start	`su - db2inst1`
	`db2jstrt 6789`

Validate	`ps -ef | grep db2jd`
Comment	DB2 Listener enables Tivoli Personalized Services Management to communicate with DB2. DB2 Listener must be started after rebooting or if it stops running.
DB2 Universal Database	
Config Program	`db2setup`
Start	`db2start`
Stop	`db2stop`

Validate	`su -tsmuser`
	`db2 connect to ispb`

Comments	The validate command initiates a connection to the Tivoli Internet Services Manager database to verify.
Everyplace Active Session Table	
Config Program	`ASTConsole`
Config File	`./conf/ASTServer.properties`
Start	`./bin/ASTServer.ksh` or `nohup ASTServer&`
Stop	Select **1** from the administration console.

Comments	This component may have its own administration console.
Everyplace Wireless Gateway	
Config Program	`wgcfg`
Start	AIX: `startsrc -s wgated` Solaris: `wgstart`
Stop	AIX: `stopsrc -s wgated` Solaris: `wgstop`
Validate	`ps -ef | grep wgated`
Comments	This component has its own administration console. Some functions can be performed through Wireless Gatekeeper. **Note**: Do not use a `kill` command to stop this component. A `kill` command may begin an unrecoverable start-stop loop of this component. If the gateway fails to stop using the above commands, issue the stop and then the `kill` command.
IBM HTTP Admin Server	
Config File	`/conf/admin.conf`

Start	`/bin/adminctl start`
	`/bin/httpd -f ./conf/admin.conf`

Stop	`/bin/adminctl stop`
IBM HTTP Server	
Config File	`/conf/httpd.conf`
Start	`/bin/apachectl start`
Stop	`/bin/apachectl stop`
Validate	`http://<hostname>/`
Comment	After rebooting a machine that has IBM HTTP Server installed, you may need to manually restart IBM HTTP Server. **Note:** On AIX, when using IBM HTTP Server with WebSphere Application Server you may not be able to restart IBM HTTP Server without rebooting the machine. See Operating system support and requirements in Requirements and prerequisites for information on how to fix this problem with an APAR.
Oracle on Solaris	
Start	`/etc/rc.oracle`
Comment	After rebooting a Solaris machine that contains Tivoli Personalized Services Manager Oracle database server, the Oracle database may need to be restarted manually.
SecureWay Directory	

Config Program	ldapxcfg dmt ldapsearch
Start	http://<hostname>/ldap
Stop	http://wes/ldap
Validate	http://<hostname>/ldap or ps -ef | grep slapd*
Comments	The validation reply, /bin/slapd -f /etc/slapd32.conf, confirms that SecureWay Directory is running. SecureWay Directory server should start automatically when the system reboots.
Tivoli Personalized Services Manager	
Start Tivoli Personalized Services Manager	Start servlets from Application Server Console: http://wes:8080 http://wes:9080 http://wes:14080/cc.html http://wes:15080 http://wes:16080 http://wes:18080/enroll.html
WebSphere Application Server Advanced Edition	
Start for the first time	/IBMWebAS/bin ./setupCmdLine.sh
Start	/IBMWebAS/bin ./startupServer.sh The log says "open for e-business" when the Application Server has started
Stop	Use admin console /bin/adminclient.sh
Validate	http://localhost/servlet/snoop
Comments	If Application Server is installed on a machine, stop the Application Server before rebooting that machine. Some functions are available through the HTTP Server administration console.
WebSphere Application Server Administrative Console	
Start	/IBMWebAS/bin ./adminclient.sh &
WebSphere Edge Server Caching Proxy (Web Traffic Express)	
Config Program	http://wes:<port>
Config File	/etc/ibmproxy.conf or ibmwesas.conf

<table>
<tr><td>Start</td><td><code>ibmproxy</code> or
<code>startsrc -s ibmproxy</code> or
<code>wslstartwte</code></td></tr>
<tr><td>Stop</td><td><code>stopsrc -s ibmproxy or</code>
<code>ps -aef|grep "ibmproxy"</code>
<code>kill ibmproxy</code></td></tr>
<tr><td>Validate</td><td><code>http://<WebTrafficExpress domain&gt;</code></td></tr>
<tr><td>Comments</td><td>This component is also controlled through Suite Manager.
WebSEAL-Lite starts with Web Traffic Express.</td></tr>
<tr><td colspan="2"><b>WebSphere Edge Server Load Balancer (Network Dispatcher)</b></td></tr>
<tr><td>Config Program</td><td><code>ndadmin</code></td></tr>
<tr><td>Start</td><td><code>ndserver start</code></td></tr>
<tr><td>Stop</td><td><code>ndserver stop</code> or
<code>ndcontrol executor stop</code></td></tr>
<tr><td>Comments</td><td>This component is also controlled through Suite Manager</td></tr>
<tr><td colspan="2"><b>WebSphere Transcoding Publisher</b></td></tr>
<tr><td>Config Program</td><td><code>SetupWizard.sh</code></td></tr>
<tr><td>Start</td><td><code>RunTranscoding.sh</code> or
<code>RunTranscoding.sh -g</code> or
<code>TransPub restart</code></td></tr>
<tr><td>Stop</td><td><code>Kill (Trans)</code></td></tr>
<tr><td>Validate</td><td><code>ps -ef | grep RunTran</code></td></tr>
<tr><td>Comments</td><td>This component may have its own administration console.</td></tr>
</table>

Default port numbers

If you are going to co-locate WebSphere Everyplace Server components on the same physical machines, you need to avoid port number assignment conflicts. The following table describes the default port numbers for the various WebSphere Everyplace Server functions. Refer to component specific documentation for information on how to change port number settings.

WebSphere Everyplace Server Functions	**Port Number**
DB2 Universal Database JDBC daemon	6789
Everyplace Active Session Table server	8017
Everyplace Wireless Gateway	
Gatekeeper and access manager	9555
Gatekeeper and access manager with SSL	9559
Gateway change password	8888
Gateway IP-LAN send/receive	8889
Gateway connection less WAP	9200
Gateway connection-oriented WAP	9201
Gateway Secure connectionless WAP	9202
Gateway Secure connection-oriented WAP	9203
IBM HTTP Server	80
SecureWay Directory	389
Secure Sockets Layer (SSL)	443
Tivoli Personalized Services Manager	
Authentication Insecure	8080
Customer Care	14080
Selfcare	15080
System Management Tool	9080
Personalization	16080
Premium	10080
Enrollment	18080
SDP Servlet	8090
Web Content Hosting	12080

WebSEAL-Lite	8081

Everyplace Suite Manager

Overview

Everyplace Suite Manager provides a centralized method for launching the administration consoles of the installed WebSphere Everyplace Server components. In addition, Suite Manager obtains information regarding the installed components and the servers where they are installed. From the Suite Manager console, you can make changes to configuration data that is stored in SecureWay Directory (LDAP).

Before you can start Everyplace Suite Manager, SecureWay Directory must be installed and running in the WebSphere Everyplace Server domain.

Suite Manager on AIX and Solaris

Installing on AIX and Solaris

To install Suite Manager on AIX and Solaris, use Everyplace Server Setup Manager. Everyplace Suite Manager is a selectable component.

Start from the desktop

After installation, you can start the console by clicking the console icon, which was created during the installation process. On AIX, the icon can be found in `/home/admin_userID/wesconsole`, `/wesconsole`, and `/usr/IBMEPS/suite/wesconsole`, and on Solaris in `/home/admin_userID/wesconsole`, `/wesconsole`, and `/opt/IBMEPS/suite/wesconsole`. Where admin_userID is the user ID specified in the Everyplace Administration Console installation.

Start from a command line

After installation, you can also start the console from the command line as follows:

1. Log on with the Administration Console user ID specified during the installation (the Everyplace Administration Console can also be started using the root userID).
2. Open a terminal window.
3. Execute the following commands:
   - **For AIX:**

     ```
     cd /usr/IBMEPS/suite
     ./wesconsole.sh
     ```
   - **For Solaris:**

     ```
     cd /opt/IBMEPS/suite
     ./wesconsole.sh
     ```

Everyplace Suite Manager ensures that all of the conditions for use, such as user privilege and prerequisite software, are correct. If the conditions are met, a list of the installed Everyplace Server components is displayed.

Suite Manager on Windows 2000

Installing on Windows 2000

To install Suite Manager on Windows:

1. Insert WebSphere Everyplace Server Disc 1 into your cdrom drive.
2. From the Start menu, select **Run...**.
3. Enter `e:\eps\win32\SuiteMgr.exe` (assuming e: is the CD drive letter).
4. Press **OK** and follow the installation program.

Start from the Start Menu

Go to **Start** > **Programs** > **WebSphere Everyplace Suite Manager** > **WebSphere Everyplace Suite Manager**.

Start from a command line

1. Open a command prompt.
2. Enter `installLocation\SuiteManager.cmd`
   Note: The default install location is c:\Program Files\SuiteManager.

Uninstalling on Windows 2000

To uninstall Suite Manager on Windows:

1. Go to **Start > Settings > Control Panel**
2. Select **Add/Remove Programs** from the menu
3. Select **WebSphere Everyplace Suite Manager**
4. Select **Remove** to begin the uninstall process.

Signing on to Everyplace Suite Manager

When you launch Everyplace Suite Manager, the logon panel is displayed. To sign on to Everyplace Suite Manager, enter the following information in the fields provided.

Field	Description	Example
LDAP Server	The name of the LDAP server where Everyplace Suite Manager is installed	joeuser.myco.com
LDAP Port	The port number for the LDAP server	389
LDAP User ID	Your LDAP User ID. (Note: This must be preceded by cn=)	cn=joeuser
LDAP Password	The password for your LDAP User ID. (Characters will appear as asterisks)	123456

You can test the server connection by clicking **Test**. You can cancel the logon by clicking **Cancel**.

Click **OK** to sign on. The Everyplace Suite Manager main panel is displayed. A hierarchical view of the WebSphere Everyplace Server domain is presented in the left pane. The default view is the system view. The right pane lists the domains running WebSphere Everyplace Server.

Getting help for Everyplace Suite Manager

WebSphere Everyplace Suite Manager contains a full on-line help system integrated into the Suite Manager console. This help is available from the Everyplace Suite Manager **Help** menu. Click on **Help topics** to launch the help system. In addition, context help is available for all Everyplace Suite Manager property dialogs. Click **Help** on the panel to display the appropriate help for the dialog.

Additional Troubleshooting

The Everyplace Suite Manager on-line help system contains a comprehensive section on troubleshooting problems involving communications disruptions between Everyplace Suite Manager and the WebSphere Everyplace Server components. The following section serves as a supplement to that information.

RMI TCP/IP Ports

In the WebSphere Everyplace Server implementation, the RMI Registry uses 1100 as its default port. However, if the default RMI Registry port is already in use, the Everyplace Server Setup Manager automatically seeks the first unoccupied port after 1100 and uses it for the RMI Registry. All scripts and LDAP settings are automatically updated to reflect this change.

If the RMI Registry must be moved from the selected port, the administrator can change the port setting by performing the following steps:

1. On the server where the port change is required, logon on as the *root* user and change the Everyplace Server Setup Manager install directory.
    - **On AIX:**

      ```
      cd /usr/IBMEPS/setup
      ```
    - **On Solaris:**

      ```
      cd /opt/IBMEPS/setup
      ```
2. Run the following command: **./WESend**.
3. Edit both the WESend and WESstart scripts by modifying the RMI_PORT environment variable assignment to the desired port number.
4. Run the following command: **./WESstart newRMI**.
5. In the LDAP directory used by WebSphere Everyplace Server, locate the Everyplace Suite Manager service for the server being changed, for example
    - serviceName=svcesp1,dc=wesserver,dc=raleigh,dc=ibm,do=com
6. The configPtr attribute of the entry contains a pointer to the eProperty where the

RMI port value must be changed, for example

  - cid=cfg1,sys=esp,sys=SDP,DC=RALEIGH,DC=IBM,DC=COM

7. Open the*settingID=RMIport* eProperty.
8. Modify the cisProperty entry with the same value as used in the scripts.

There are no settings to change on the Everyplace Suite Manager console. However, the server that has been changed must be refreshed on the Everyplace Suite Manager console before resuming remote operations.

The RMI daemon port default is 1098. If the RMI daemon port must be changed, perform the following:

1. As the *root* user, change to the Everyplace Server Setup Manager install directory.
   - **On AIX:**

     ```
     cd/usr/IBMEPS/setup
     ```
   - **On Solaris:**

     ```
     cd/opt/IBMEPS/setup
     ```
2. Run the following command:**./WESend**
3. Edit both the WESend and WESstart scripts by modifying the RMID_PORT environment variable assignment to the desired port number.
4. Run the following command: **./WESStart newRMI**

There are no required changes to LDAP.

Uninstalling the WebSphere Everyplace Server components

The Uninstall Manager is part of Everyplace Suite Manager and allows you to uninstall WebSphere Everyplace Server components that are running on a local server where Everyplace Suite Manager is installed. The Uninstall Manager provides a list of components that you can select for removal. Some components, such as DB2 Universal Database, WebSphere Application Server or IBM HTTP Server, will not appear on the list. These components are automatically removed when they are no longer used by other WebSphere Everyplace Server components.

The Uninstall Manager uses SecureWay Directory to determine component usage, so SecureWay Directory must be available while the Uninstall Manager is running. If the Uninstall Manager is run without SecureWay Directory, a system inconsistency may occur.

Configuring the Uninstall Manager

The Uninstall Manager stores its configuration values in the `{instdir}/conf/IBMEPS.properties` file. The following properties are configurable for the Uninstall Manager:

Uninstall Manager properties

Property	Definition	Default value
Ldap.server	Full name of LDAP server	
Ldap.port	LDAP protocol port number	389
Ldap.userid	User ID used to connect to LDAP	
Ldap.password	Password used to connect to LDAP, encrypted	
uninstall.log.opts	Options for `everyplace_admin.log` -- append/new	append

uninstall.trace.opts	Options for `everyplace_admin.trace` -- append/new	new
uninstall.debug.level	Debugging level values 0-5 (NONE-MAX)	0

All of the properties are available through the Setup Manager **Properties** panel. In addition, the properties can be updated through Everyplace Suite Manager by selecting the following:

1. **Suite Manager Properties** and then **General** from the **File** menu.
2. **Properties** and then **Uninstall** from the **File** menu.

Starting the Uninstallation program

From the command line

To start the Uninstall Manager from the command line, enter the following:

- For AIX:

  ```
  /usr/IBMEPS/setup/uninstall.sh
  ```

- For Solaris:

  ```
  /opt/IBMEPSIn/setup/uninstall.sh
  ```

Uninstallation process

The Uninstall Manager provides Java-based wizards to assist with component removal. These wizards include a panel to select components for removal, a summary panel, a status bar, and a completion message.

To begin the uninstallation process:

1. From the components pane, select the component you want to uninstall by highlighting it. Selection of multiple components and subcomponents is allowed.
2. A list of related subcomponents is displayed in the subcomponents pane.
   - If you clicked on the check box when you highlighted the component, all of the related subcomponents will be selected. You can deselect any of the subcomponents by clicking on their corresponding check boxes.
   - If you did not click on the check box when you highlighted the component, the related subcomponents will be displayed but are not selected. You can select the subcomponents you want to uninstall by clicking on their corresponding check boxes.
3. The description field at the bottom of the panel will contain the component's or

subcomponent's installation status based on the local installation check and installation record stored in the WebSphere Everyplace Server installation database on the SecureWay directory server.

4. Click **Next**. Component dependency checks are performed. If no dependencies are found, the **Uninstallation Summary** panel is displayed. Otherwise, if dependencies are found, a warning message will appear.
   - Click **Yes** to force the uninstall for the selected component.
   - Click **No** to cancel the uninstall. You are returned to the previous panel where you can modify your selections.

After uninstallation is complete, you can view a log file to check the result of uninstallation for each component or sub-component. The log file is `everyplace_uninstall.log`, as specified in the `ibmeps.properties` file. The default location is `{instdir}/logs`.

Uninstall Manager only removes files that were installed by Setup Manager. No customer data is removed.

Uninstallation Summary Panel

The **Uninstallation Summary** panel shows a list of the WebSphere Everyplace Server components and sub-components you selected to uninstall. If you agree with the list, click the **Uninstall** button. A message will appear, stating there is no chance to halt the uninstall process after it begins and asking if you want to continue. Click the **yes** button to begin the uninstallation. Click the **no** button, to return to the **Uninstallation Summary** panel.

Uninstallation status bar

The **Uninstallation Status** bar on the **Uninstallation Summary** panel shows the progress of the uninstallation. The phrase **{Complete}** will appear next to each component as it is successfully uninstalled.

If you selected to uninstall all of the WebSphere Everyplace Server components, the Everyplace Server Package is listed at the end of the list of components. If the uninstallation of any component or subcomponent fails, the Everyplace Server Package uninstallation is skipped so that the Uninstall Manager remains installed.

Uninstallation completion

The **Uninstallation Completion** panel will appear when all of the selected WebSphere Everyplace Server components and subcomponents are successfully uninstalled. Click the **View logfile** button to display the uninstallation log. Click the **Finish** button to terminate the Uninstall Manager.

Uninstall component and subcomponent dependencies

You can uninstall the following components at any time by selecting them from the uninstallation **Select Components** panel:

- Everyplace Intelligent Notification Services
- Everyplace Location Based Services
- Everyplace Suite Manager
- Everyplace Synchronization Manager
- MQSeries Everyplace
- WebSEAL-Lite
- WebSphere Edge Server Load Balancer (Network Dispatcher)
- WebSphere Transcoding Publisher

The following WebSphere Everyplace Server components have dependencies that will affect uninstallation:

- Everyplace Active Session Table
- Everyplace Wireless Gateway
- Tivoli Personalized Services Manager
- SecureWay Directory
- WebSphere Edge Server Caching Proxy (Web Traffic Express)

Everyplace Active Session Table

A secondary AST server can be uninstalled at any time.

A primary AST server cannot be uninstalled if a secondary AST server exists.

If the selected AST table is uninstallable, a warning message appears indicating that a WebSEAL-Lite server or Everyplace Wireless Gateway server may be using the AST server. If a WebSEAL-Lite server or Everyplace Wireless Gateway server is found in the domain, you will not be permitted to uninstall the AST server.

Everyplace Wireless Gateway

You can uninstall the following Everyplace Wireless Gateway files at any time:

- `Ardis Support`
- `Dial Support`
- `Gatekeeper`

- `IP LAN Support`
- `Mobitex Support`
- `Motorola PMR Support`
- `Modacom-SCR Support`
- `DataTAC Support`
- `X.25 Support`

If you select to uninstall `Gateway`, and if any of the Everyplace Wireless Gateway files listed above have not been selected for uninstallation, you will receive a message prompting you to select those files. Click the **OK** button to return to the **Select Component** panel.

If the `Gateway` file becomes uninstallable, and any `Everyplace Wireless Gateway Services` file is left in the WebSphere Everyplace Server domain, a warning message appears. Click the **Yes** button to enforce the `Gateway` uninstallation.

If you uninstall Everyplace Wireless Gateway and reinstall it on the same server, you may get a message indicating that "the server exists" during Gatekeeper configuration. Disregard this message. The Gatekeeper configuration will perform correctly because the existing resource in the LDAP directory is reused.

Tivoli Personalized Services Manager

You must uninstall Tivoli Personalized Services Manager files before you uninstall the IBM HTTP Server. The Uninstall Manager will not uninstall the IBM HTTP Server until you have uninstalled all of the Tivoli Personalized Services Manager components, excluding Tivoli Personalized Services Manager Database Integration. You can select to uninstall the Tivoli Personalized Services Manager Database Integration separately from the other components.

If the AST Server is running on the machine in which Tivoli Personalized Services Manager is installed, you must stop it before refreshing or removing Tivoli Personalized Services Manager.

If you are uninstalling the Tivoli Personalized Services Manager Database Integration, you must first verify that there are no active connections to the Tivoli Personalized Services Manager Database. If there are any active connections, deactivate them before attempting to remove the Database Integration.

The following Tivoli Personalized Services Manager components can be removed from the system:

- Customer Care
- Enrollment Server
- Self Care
- Portal Toolkit

- System Management
- Everyplace Server Enabler
- Tivoli Database Integration
- Tivoli Device Manager

With the Uninstall Manager, you may remove all the Tivoli Personalized Services Manager components or none of them. You cannot remove subcomponents at this time. If you would like more control over the Tivoli Personalized Services Manager uninstall, refer to the Tivoli Personalized Services Manager Readme file.

Running the Tivoli Personalized Services Manager uninstallation

In order to run the Tivoli Personalized Services Manager uninstallation for both Database Integration and the components the following steps must be taken on both AIX and Solaris:

1. Verify that your Java version is 1.2 or higher using the following commands:
   `java -version`
2. If this command returns a version less than 1.2 then you need to update the link in `/usr/bin` to point to a Java version greater than 1.2. To update the link enter the following commands:
   ```
   rm /usr/bin/Java
   ln -s <Your Java 1.2 binary> /usr/bin/Java
   ```
   Example for AIX JRE 1.2: `ln -s /usr/java_dev2/jre/sh/Java /usr/bin/Java`
3. Run the installation program again.

Uninstall Tivoli Personalized Services Manager servlets from WebSphere Application Server

Any modified servlets must be manually removed before uninstalling Tivoli Personalized Services Manager.

1. Stop and remove the servlets from the WebSphere Application Server Administration Console.
2. Remove all the Tivoli Personalized Services Manager servlet entries from the UNIX registration using smitty on AIX or admintool on Solaris.
3. Remove the following directories and all files within them:
   - /usr/TivTSM
   - /usr/TivDMS
   - /usr/TivDMSUnInst

   Be sure to substitute /opt for /usr if you are removing the files from Solaris.
4. If you have not already done so, remove the /jakarta-tomcat-3.2.1 directory and all files within it.

SecureWay Directory

If you select to uninstall SecureWay Directory, and any components are left in the WebSphere Everyplace Server domain or on the server performing the uninstallation, a warning message appears. Click the **Yes** button to enforce the SecureWay Directory uninstallation.

WebSphere Edge Server Caching Proxy

The `Everyplace Wireless Gateway Services` file can be uninstalled at any time.

If you select to uninstall Caching Proxy, and any of the `Content Based Routing` or `WebSEAL-Lite` files exist on the server and have not been selected for uninstallation, or if any of the WebSphere Transcoding Publisher files remain in the WebSphere Everyplace Server domain, a warning message appears. Click the **Yes** button to enforce the Caching Proxy uninstallation.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
DB2
DB2 Universal Database
IBM
SecureWay
MQSeries
REDBOOKS
RS/6000
ThinkPad
Tivoli
WebSphere
WorkPad

The following terms are trademarks of other companies:

Sun, Sun Microsystems, all Sun-based trademarks and logos, Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Incorporated.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Domino, Lotus, and Lotus Notes are trademarks of the Lotus Development Corporation.

Netscape, Netscape Navigator, Netscape Communicator are registered trademarks of Netscape Communications Corporation in the United States and other countries.

Microsoft, Windows, Windows NT, Windows 98, the Windows 95 logo, and/or other Microsoft products referenced herein are either trademarks or registered trademarks of Microsoft Corporation.

i-mode is a registered trademark of NTTDoCoMo.

Other company, product, and service names may be trademarks or services marks of others.